

# PATENT ABSTRACTS OF JAPAN

(11)Publication number : 2003-173301

(43)Date of publication of application : 20.06.2003

(51)Int.Cl.

G06F 13/00

G06F 15/00

H04L 12/24

(21)Application number : 2001-373522

(71)Applicant : HITACHI LTD

(22)Date of filing : 07.12.2001

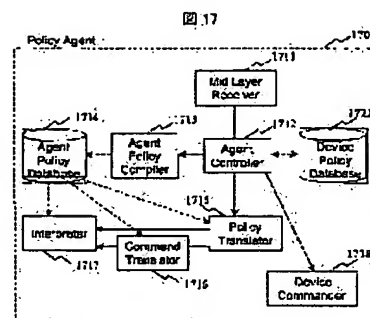
(72)Inventor : KANEDA YASUSHI  
AOKI ATSUSHI

## (54) NETWORK, SERVER AND POLICY SERVER OF STORAGE

### (57)Abstract:

**PROBLEM TO BE SOLVED:** To realize a policy system capable of easily performing the addition or deletion of a policy as occasion demands by reusing an existing interface, and also applying the policy to new equipment having a new equipment interface or command by a short-time development in a site as occasion demands.

**SOLUTION:** A policy manager registers a policy kind definition in a database in the definition thereof, retrieves the policy kind definition in the input of a policy, and inspects the adaptability of the inputted policy in reference to it. A policy agent registers a definition of conversion rule of policy in the definition thereof, retrieves the conversion rule in the input of the policy, and converts the policy into a command train by using it.



## LEGAL STATUS

[Date of request for examination]

01.12.2004

[Date of sending the examiner's decision of rejection]

[Kind of final disposal of application other than the examiner's decision of rejection or application converted registration]

[Date of final disposal for application]

[Patent number]

[Date of registration]

[Number of appeal against examiner's decision of rejection]

[Date of requesting appeal against examiner's decision of rejection]

[Date of extinction of right]

Copyright (C); 1998,2003 Japan Patent Office

(19) 日本国特許庁 (J P)

(12) 公開特許公報 (A)

(11) 特許出願公開番号  
**特開2003-173301**  
(P 2 0 0 3 - 1 7 3 3 0 1 A)  
(43) 公開日 平成15年6月20日 (2003. 6. 20)

(51) Int. Cl. 7	識別記号	F I	テーマコード (参考)		
G06F 13/00	353	G06F 13/00	353	B	5B085
15/00	310	15/00	310	B	5B089
H04L 12/24		H04L 12/24			5K030

審査請求 未請求 請求項の数16 O L (全24頁)

(21) 出願番号 特願2001-373522 (P 2001-373522)

(22) 出願日 平成13年12月7日 (2001. 12. 7)

(71) 出願人 000005108  
株式会社日立製作所  
東京都千代田区神田駿河台四丁目6番地  
(72) 発明者 金田 泰  
東京都国分寺市東恋ヶ窪一丁目280番地  
株式会社日立製作所中央研究所内  
(72) 発明者 青木 篤  
神奈川県横浜市戸塚区戸塚町5030番地 株  
式会社日立製作所ソフトウェア事業部内  
(74) 代理人 100075096  
弁理士 作田 康夫

最終頁に続く

(54) 【発明の名称】 ネットワーク、サーバおよびストレージのポリシーサーバ

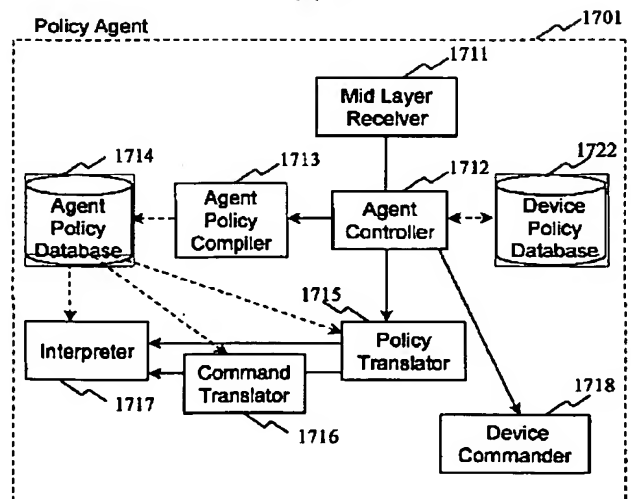
(57) 【要約】

【目的】 本発明の目的は、既存のインタフェースを再利用して容易に必要なに応じてポリシーの追加や削除ができ、あらたな機器インタフェースやコマンドをもつあらたな機器に、現場における短時間の開発によってポリシーを必要に応じて適用できるポリシーシステムを実現することである。

【構成】 ポリシーマネージャがポリシー種類定義時に当該定義をデータベースに登録し、ポリシー入力時に前記のポリシー種類定義を検索し参照して入力ポリシーの妥当性を検査する。ポリシーエージェントがポリシーの変換規則定義時に当該定義をデータベースに登録し、ポリシー入力時に前記の変換規則を検索し使用してコマンド列に変換する。

【効果】 既存のインタフェースを再利用して容易に必要なに応じてポリシーの追加や削除ができ、あらたな機器インタフェースやコマンドをもつあらたな機器に、現場における短時間の開発によってポリシーを必要に応じて適用できるようにすることができる。

図 17



## 【特許請求の範囲】

【請求項 1】ネットワークに接続された機器をポリシーを使用して制御するポリシーサーバにおいて、前記のポリシーサーバは特定の種類のポリシーを前記の機器のコマンド列に変換する方法を記述したデータを入力して保管し、前記の特定の種類の属する第 1 のポリシーが入力されたときに保管されていた前記の変換方法を使用して前記の第 1 のポリシーをコマンド列に変換して前記の機器に送付することを特徴とするポリシーサーバ。

【請求項 2】請求項 1 のポリシーサーバにおいて、前記の変換方法がコマンドのテンプレートと、前記のテンプレートの可変部分をうめるデータ生成手段とによって記述されることを特徴とするポリシーサーバ。

【請求項 3】請求項 2 のポリシーサーバにおいて、前記の機器がコマンドライン・インタフェースを具備しているときに、前記のテンプレートとして前記のコマンドライン・インタフェースにおけるコマンドの文字列テンプレートを使用し、前記のデータ生成手段が文字列を生成することを特徴とするポリシーサーバ。

【請求項 4】請求項 2 のポリシーサーバにおいて、前記の機器がプロトコル SNMP によって設定される管理情報ベースを具備しているときに、前記のテンプレートが SNMP による管理情報ベースへの設定コマンドのテンプレートであり、前記のデータ生成手段が前記の設定コマンドの引数をうめることを特徴とするポリシーサーバ。

【請求項 5】請求項 2 のポリシーサーバにおいて、前記の機器がプロトコル COPS-PR によって設定されるポリシー情報ベースを具備しているときに、前記のテンプレートが COPS-PR によるポリシー情報ベースへの設定コマンドのテンプレートであり、前記のデータ生成手段が前記の設定コマンドの引数をうめることを特徴とするポリシーサーバ。

【請求項 6】請求項 2 のポリシーサーバにおいて、前記の機器がハードウェア・テーブルへのかきこみによって設定されるときに、前記のテンプレートが前記のハードウェア・テーブルへのかきこみ位置およびデータ幅を指示するデータであり、前記のデータ生成手段が前記の位置およびデータ幅にかきこむ値を生成することを特徴とするポリシーサーバ。

【請求項 7】ネットワークに接続された機器をポリシーを使用して制御するポリシーシステムにおいて、前記のポリシーシステムが機器独立なポリシーを管理するポリシーマネージャと、機器独立なポリシーを機器依存の設定に変換するポリシーエージェントとで構成されているときに、前記のポリシーマネージャは特定の種類のポリシーを前記の機器のコマンド列に変換する方法を記述した第 1 のポリシーを前記のポリシーエージェントに配布することが指示されると前記の第 1 のポリシーをふくむポリシー適用要求を前記のポリシーエージェントに送付し、前記のポリシーエージェントは前記の第 1 のポリ

シーを保管し、前記の特定の種類の属する第 2 のポリシーが入力されたときに保管されていた前記の変換方法を使用して前記の第 2 のポリシーをコマンド列に変換して前記の機器に送付することを特徴とするポリシーシステム。

【請求項 8】請求項 7 のポリシーサーバにおいて、前記のポリシーマネージャは前記の第 1 のポリシーを前記のポリシーエージェントから削除することが指示されると前記の第 1 のポリシーの名称をふくむポリシー削除要求を前記のポリシーエージェントに送付し、前記の削除要求にしたがって前記のポリシーエージェントは保管されていた前記の第 1 のポリシーを削除することを特徴とするポリシーシステム。

【請求項 9】ネットワークに接続された機器をポリシーを使用して制御するポリシーサーバにおいて、前記のポリシーサーバは特定の種類の属するポリシーの定義を記述したデータを入力して保管し、前記の特定の種類の第 1 のポリシーが入力されたときに保管されていた前記の定義を使用して前記の第 1 のポリシーが前記の定義に合致しないときエラーメッセージを出力することを特徴とするポリシーサーバ。

【請求項 10】ネットワークに接続された機器をポリシーを使用して制御するポリシーサーバにおいて、前記のポリシーサーバはポリシークラスの定義を記述した前記の第 1 のポリシーを入力して保管し、ユーザ・インタフェースから第 1 のポリシーの前記のポリシーサーバへの配布が指示されると、前記のユーザ・インタフェースに表示されたポリシークラスのリストに前記の第 1 のポリシーに指定された前記のポリシークラスの名称を追加することを特徴とするポリシーサーバ。

【請求項 11】請求項 10 のポリシーサーバにおいて、ユーザ・インタフェースから第 1 のポリシーの前記のポリシーサーバからの削除が指示されると、前記のユーザ・インタフェースに表示されたポリシークラスのリストから前記のポリシークラスの名称を削除することを特徴とするポリシーサーバ。

【請求項 12】ネットワークに接続された機器を変数と値の対のならばの形式を使用して記述されたポリシーを使用して制御するポリシーサーバにおいて、前記のポリシーサーバは特定の種類のポリシーに出現可能な変数名のリストを記述したデータを入力して保管し、前記の特定の種類の第 1 のポリシーが入力されたときに保管されていた前記のリストを使用して前記の第 1 のポリシーに前記のリストにふくまれない変数が出現するときにエラーメッセージを出力することを特徴とするポリシーサーバ。

【請求項 13】ネットワークに接続された機器を変数と値の対のならばの形式を使用して記述されたポリシーを使用して制御するポリシーサーバにおいて、前記のポリシーサーバは特定の名前の変数に対して代入可能な値の

型のリストを記述したデータを入力して保管し、第 1 のポリシーが入力されたときに保管されていた前記のリストを使用して前記の第 1 のポリシーにおいて前記のリストにふくまれない型の値が前記の名前の変数に代入されているときにエラーメッセージを出力することを特徴とするポリシーサーバ。

【請求項 14】請求項 1 のポリシーサーバにおいて、前記のポリシーサーバが式を評価するインタプリタを内蔵し、前記の変換方法がコマンドのテンプレートと、演算子または関数およびだしをふくむ式の形式のデータ生成手段とによって記述され、前記のデータ生成手段が前記のインタプリタによって評価された結果によって前記のテンプレートの可変部分をうめることによりコマンドを生成することを特徴とするポリシーサーバ。

【請求項 15】請求項 1 のポリシーサーバにおいて、前記の変換方法がコマンドのテンプレートと、引数として 2 個の整数を指定した関数およびだし形式のデータ生成手段とによって記述され、前記の変換方法にしたがってコマンドが生成されるごとに前記のデータ生成手段は前記の第 1 と第 2 の整数のあいだの整数値によって前記のテンプレートの可変部分をうめることにより前記のコマンドを生成し、前記の第 1 と第 2 の整数のあいだの整数値がつくされるとエラーメッセージを出力することを特徴とするポリシーサーバ。

【請求項 16】ネットワークに接続された第 1 の機器と第 2 の機器とをポリシーを使用して制御するポリシーサーバにおいて、前記のポリシーサーバは特定の種類のポリシーを前記の第 1 の機器のコマンド列に変換する第 1 の方法を記述したデータを入力して保管するとともに前記の第 2 の機器のコマンド列に変換する第 2 の方法を記述したデータを入力して保管し、前記の特定の種類の属する第 1 のポリシーが入力されたときに保管されていた前記の第 1 の方法を使用して前記の第 1 のポリシーをコマンド列に変換して前記の第 1 の機器に送付するとともに前記の第 2 の方法を使用して前記の第 1 のポリシーをコマンド列に変換して前記の第 2 の機器に送付することを特徴とするポリシーサーバ。

【発明の詳細な説明】

【0001】

【発明の属する技術分野】本発明は、LAN スイッチ等のネットワーク・ノードによって構成されたネットワークと、Web サーバ等のサーバと、SAN、NAS 等のストレージの制御に関する。

【0002】

【従来の技術】本発明に関連する第 1 の従来技術としてネットワークのポリシー制御方式がある。ネットワークのポリシー制御方式に関しては IETF (Internet Engineering Task Force) 等において議論されているが、その概説としてはつぎの文献がある。

【0003】製品化始まったポリシー・サーバ、日経

インターネットテクノロジー、1999 年 6 月号、144 ページから 151 ページポリシー制御のないネットワーク・システムにおいては、ネットワーク機器の QoS 管理機能 (サービス品質管理機能) やセキュリティ管理機能などを制御する際に、ネットワーク機器ごとに設定をおこなう必要がある。しかし、ポリシー制御ネットワーク・システムにおいては、ポリシーサーバとよばれるコンピュータに設定の方針すなわちポリシーを指定することによって、少量の情報を入力するだけでネットワーク全体への設定ができるようにしている。また、時刻をこまかく指定してポリシーを変更したり、アプリケーション・プログラムからの要求に応じて動的にポリシーを変更するなど、人間のオペレータでは実現困難なネットワーク制御を可能にしている。

【0004】ポリシーは通常、ポリシー規則とよばれる規則のならびとして記述される。ポリシー規則は条件-動作型の規則である。すなわち、ある条件がなりたつときにとるべき動作を記述する規則である。1 個のポリシーのなかに任意の条件や動作をふくんだポリシーを許容することも可能だが、ポリシーの処理やユーザ・インタフェースの単純化などのために、特定の形式や内容をもつポリシー規則だけをふくむように制約することもできる。Hewlett-Packard 社および日立製作所の製品であるポリシーサーバ PolicyXpert (TM) においては、ポリシー規則において指定できる動作に型の概念を導入し、1 個のポリシーには特定の型のポリシー規則だけがふくまれるようにしている。そのため、各ポリシーにもそれがふくむことができる動作の型に対応する型 (ポリシー型) をもっている。PolicyXpert のポリシーの形式や意味についてはつぎの文献に記述されている。

【0005】HP OpenView PolicyXpert User's Guide, J1360-90010 (<http://ovweb.external.hp.com/ovnsmdps/pdf/j1360-90010.pdf>), Hewlett-Packard, 2001 年。ポリシーの形式や意味については、IETF (Internet Engineering Task Force) において「ポリシー情報モデル」というかたちで標準化がおこなわれている。核となる情報モデルとしてはつぎの文書がある。

【0006】Moore, B., Elleson, E., Strassner, J., and Westerinen, A., "Policy Framework Core Information Model Version 1 Specification", RFC 3060 (<http://www.ietf.org/rfc/rfc3060.txt>), IETF, 2001 年。Moore, B., Rafalow, L., Ramberg, Y., Snir, Y., Strassner, J., Westerinen, A., Chadha, R., Brunner, M., Cohen, R., "Policy Core Information Model Extensions", draft-ietf-policy-pcim-ext-05.txt (<http://www.ietf.org/internet-drafts/draft-ietf-policy-pcim-ext-05.txt>), Internet Draft, IETF, 2001 年。ポリシーを機器にダウンロードする際のプロトコルには複数の候補があるが、代表的なものとして COPS (Common Open Policy Service) プロトコルがある。COPS プロト

コルは IETF において、つぎの文献によって提案されている。

【0 0 0 7】D. Durham, 編, The COPS (Common Open Policy Service) Protocol, RFC 2748 (<http://www.ietf.org/rfc/rfc2748.txt>), IETF, 2000 年. F. Reichmeyer 他著, COPS Usage for Policy Provisioning (COPS-PR), RFC3084 (<http://www.ietf.org/rfc/rfc3084.txt>), IETF, 2001 年. また, ダウンロードする際のポリシーの表現に関しては, PIB (Policy Information Base) が提案されている. その一例として, つぎの文献に記述されたものがある。

【0 0 0 8】M. Fine 他著, Quality of Service Policy Information Base, draft-mfine-cops-pib-05.txt (<http://www.ietf.org/internet-drafts/draft-mfine-cops-pib-05.txt>), インターネット・ドラフト, IETF, 2001 年. 本発明に関連する第 2 の従来技術として, インターネット上での QoS (Quality of Service) すなわちサービス品質を保証するための技術がある. 代表的な QoS 保証技術としては Differentiated Services 技術 (以下 DiffServ 技術とよぶ) がある. DiffServ 技術に

関してはつぎのような文献がある。

【0 0 0 9】S. Blake 他著, An Architecture for Differentiated Services, RFC 2475 (<http://www.ietf.org/rfc/rfc2475.txt>), IETF, 1998 年. K. Nichols 他著, A Two-bit Differentiated Services Architecture for the Internet, RFC 2638 (<http://www.ietf.org/rfc/rfc2638.txt>), IETF, 1999 年. DiffServ 技術においては, 第 1 のネットワーク・アプリケーションからネットワークを介して第 2 のネットワーク・アプリケーションとの間で一連のパケットが通信されるとき, これらが 1 個の「フロー」すなわち一連のパケットの流れに属していると考えられる. ある IP パケットがあるフローに属しているかどうかは, IP パケット上の始点と終点の IP アドレス, プロトコル, さらにそのプロトコルが TCP または UDP であるときはそのポートを識別することによって判定することができる。

【0 0 1 0】第 1 のネットワーク・アプリケーションから第 2 のネットワーク・アプリケーションへのパス上には, まずネットワークへの入口のエッジ・ルータがあり, 0 個またはそれ以上のコア・ルータがあって, ネットワークからの出口のエッジ・ルータがある. このとき, DiffServ 技術においては, 入口のエッジ・ルータにおいて複数のフローをまとめてパケット上の DS フィールド (Differentiated Services フィールド) に特定の値によってマーキングし, それ以降はその値をもつパケットをまとめてひとつのフロー (集成フロー aggregated flow と呼ぶ) としてあつかう. DS フィールドにふくまれる値は DSCP (Differentiated Services Code Point) とよばれる. 集成フローをつくることによって, コア・ルータにおいては, DSCP だけを判定することによ

って集成フローごとに帯域幅やパケット転送の優先度などの QoS 条件を制御することができる. DiffServ 技術を使用することによって, フローを集成し DSCP だけで判定できるようになり, QoS 条件の制御のためのコア・ルータの負荷を軽減することができる。

【0 0 1 1】Diffserv 技術を使用することにより, Ethernet による LAN, IP ネットによる WAN など, 複数のネットワークが接続された環境においても, end-to-end で通信品質を保証することができる. なぜなら, これらのネットワークにおいてはフローの識別や優先度の制御などをおなじようにおこなうことができるからである。

【0 0 1 2】

【発明が解決しようとする課題】従来は独立に開発され使用されてきたネットワーク, サーバすなわちワークステーションやパソコン, ディスク・ストレージを統合的に管理する必要がたかまってきた。たとえば, 企業においても LAN, 事業所間をむすぶ WAN, Web サーバやデータベース・サーバと, それらが使用するストレージを統合的にポリシー制御することによって, それらをより容易に戦略的に利用することができるようになる。

【0 0 1 3】このような環境においては, 第 1 に, 利用可能な最新技術を導入後ただちにポリシー制御できるようにし, ニーズにもとづいてポリシーを設計・使用することができることがもとめられる. 前記のような IETF におけるポリシーや QoS の標準化を待っていることはできない. ところが, 従来のポリシーシステムにおいては, あらかじめポリシーサーバやポリシーエージェントにくみこまれた特定の機能をもつポリシーしかつかうことができなかった。

【0 0 1 4】上記のような環境においては, 第 2 に, 1 個のポリシーをネットワーク機器, サーバ, ストレージなどに配布することによって, これらを統合的に管理できるようにすることがもとめられる. しかし, 従来のポリシーシステムにおいては, あらかじめ定められた種類の機器インタフェース, たとえば SNMP と MIB とのくみあわせや COPS-PR と PIB のくみあわせ, 特定のコマンドライン・インタフェース (CLI), 特定の API などしか使用することができず, これらのインタフェースをもっていない機器をあつかうことができなかった. そのため, 標準化が比較的すすんでいるネットワークにおいてもかぎられた機器しか制御することができず, 標準化がすすんでいないサーバやストレージにおいてはポリシー制御の可能性はきわめてかぎられていた。

【0 0 1 5】本発明の第 1 の目的は, 上記の第 1 の問題を解決するために, 必要に応じてポリシーの追加ができるようにすることである. しかも, そのためにユーザを特別に訓練する必要がある汎用プログラミング言語や新規のインタフェースを使用するのではなく, 既存のインタフェースを再利用して容易にできるようにすること

ある。本発明の第 2 の目的は、上記の第 2 の問題を解決するために、あらたな機器インタフェースやコマンドをもつあらたな機器に、現場における短時間の開発によってポリシーを必要に応じて適用できるようにすることである。

【 0 0 1 6 】

【課題を解決するための手段】前記の 2 個の課題は、ポリシーサーバおよびそのプロキシー内につぎのような 2 個の手段をそなえることによって解決することができる。第 1 の手段は、ポリシーの形式を使用して記述された機器独立なポリシークラスすなわちポリシーの種類の追加または削除が指示されたとき前記のポリシークラス定義を第 1 のデータベースに格納するかまたは第 1 のデータベースから削除し、前記のポリシークラス定義によって定義された種類のポリシーが入力されたとき前記のポリシークラス定義を前記の第 1 のデータベースからとりだして参照することによって前記のポリシーの妥当性を検査するポリシーマネージャである。第 2 の手段は、ポリシーの形式を使用して記述された特定の種類のポリシーからコマンド列への機器依存の変換規則の追加または削除が指示されたとき前記の変換規則を第 2 のデータベースに格納するかまたは第 2 のデータベースから削除し、前記の種類のポリシーが入力されたとき前記の変換規則を前記の第 2 のデータベースからとりだして使用することによってコマンド列に変換するポリシーエージェントである。

【 0 0 1 7 】前記の第 1 の手段を使用することによって、あらたな機能をもつポリシークラスを追加し、前記のポリシークラスに属するポリシーの妥当性を検査することができるようになる。また前記の第 2 の手段を使用することによって特定のポリシークラスに属するポリシーを機器に設定することができるようになる。したがって、必要に応じて新機能をもつポリシークラスを追加・削除することが可能になり、本発明の第 1 の目的を達することができる。また、必要に応じてあらたな機器インタフェースやコマンドをもつ機器に特定のポリシークラスに属するポリシーを機器に適用することが可能になり、本発明の第 2 の目的を達することができる。

【 0 0 1 8 】

【発明の実施の形態】本発明の実施例を説明する。この実施例は仮想的なコンピュータ・システム開発・販売会社である X 社のシステムに関する。X 社は顧客の情報を詳細にあつめてデータベース化し、Web をつうじて社内にもその情報を提供している。X 社には製造部門と販売部門とがあるが、販売部門においては毎日朝 9:00 から 10:00 まで、セールスマンが前日にあつめた情報を集中的にデータベースに入力する。そのため、9:00 から 10:00 までは社内のネットワーク、サーバ、ストレージ資源を販売部門に優先的にわりあてるひとつのポリシーを記述したい。具体的には、この時間帯に、ネットワー

ク (LAN) においては販売部門からのトラフィックに他の部門よりたかい優先度をあたえたい。また、サーバに関しては、サーバの各リソースのなかから 50% の部分が使用できるようにしたい。さらに、ストレージ (SAN および NAS) に関しては当該のデータのキャッシングを指定することにより、効率的なアクセスを可能にしたい。これらをひとつのポリシーとして記述することによって、統一的にあつかえるようにしたい。

【 0 0 1 9 】この実施例におけるネットワークおよび機器の構成を図 1 を使用して説明する。LAN (Local Area Network) 121 にはクライアント 110 およびクライアント 111 が接続されている。クライアント 110 およびクライアント 111 は PC、ハンドヘルド・コンピュータ、または携帯電話である。LAN 121 には LAN スイッチ 122 がふくまれているが、LAN スイッチ 122 はサーバ 131 およびサーバ 132 に接続されている。サーバ 131 およびサーバ 132 は LAN 121 に接続されると同時に SAN (Storage Area Network) 141 にも接続されている。SAN 141 にはストレージ機器 151、ストレージ機器 152 が接続されるとともに、ストレージ機器 151、ストレージ機器 152 を制御するストレージ・コントローラ 153 も接続されている。ストレージ・コントローラ 153 は LAN 121 を経由した管理のため、LAN 121 にも接続されている。また、LAN 121 にも NAS (Network Attached Storage) 機器 150 が接続されている。LAN 121 およびストレージ・コントローラ 153 経由で SAN 141 に接続された各機器を統合的に制御するため、ポリシーサーバ 112 が LAN 121 に接続されている。

【 0 0 2 0 】つぎに、図 2 を使用してポリシーサーバ 112 の構成を説明する。ポリシーサーバ 112 はパーソナル・コンピュータまたはワークステーションのような汎用のコンピュータを使用して実装される。図 2 を構成する要素のなかでオペレータ・コンソールはディスプレイ装置とキーボードおよびマウスという入出力機器とソフトウェアとによって構成されるが、これ以外の部分はすべてソフトウェアによって実現される。また、ポリシーリポジトリ 213 はハードディスク上または主記憶上におかれる。

【 0 0 2 1 】オペレータ・コンソール 211 はオペレータからの入力を受けつけ、またオペレータへの出力をおこなう。オペレータ・コンソール 201 を使用してポリシーやポリシー規則が追加、削除または更新されるが、このようなオペレータ・コンソール 211 の入出力はポリシーマネージャ 212 によって制御される。入力されたポリシーはポリシーリポジトリ 213 に保存されると同時に、ネットワーク・エージェント 215 によって LAN スイッチ 122 に適用され、サーバ・エージェント 216 によってサーバ 131 および 132 に適用される。また、ストレージ・エージェント 217 によってストレージ 150 に適用され、またストレージ・コントローラ 153 を

経由してストレージ 151、152 に適用される。ここでサーバ・エージェント 216 およびストレージ・エージェント 217 がポリシーサーバ 121 内にあるのに対してネットワーク・エージェント 215 がポリシーサーバ 121 外にあるのは、ネットワーク・エージェント 215 は LAN スイッチ 121 のハードウェア・テーブルをアクセスするため、LAN スイッチ 121 内にうめこまれるからである。

【0022】つづいて、図 3 を使用してオペレータ・コンソール 211 の主画面 301 について説明する。主画面 301 はポリシークラス、ポリシーおよびポリシー規則の表示、定義、削除などを指示するための画面である。ここでポリシークラスとはつぎのような性質をもつポリシーのあつまりを意味する：あるクラスに属するポリシーはあるきまった範囲の機能をもっていて、使用できる条件や動作の範囲がきめられている。すなわち、ポリシークラスは PolicyXpert におけるポリシー型にちかい概念である。

【0023】主画面 301 は 3 個のフレーム 311、321、331 によって構成されている。ここで、フレームとは、Web ブラウザにおいてウィンドウを分割してそれぞれにことなる Web ページを表示する際に使用するフレームにちかい。クラス・フレーム 311 にはクラスリスト 312、Show ボタン 313、Delete ボタン 314、Insert ボタン 315、テキストフィールド 316 がふくまれている。クラスリスト 312 にはポリシークラス名のリストが表示されるが、マウスを使用してポリシークラスを選択することができる。Show ボタン 313 は選択したポリシークラスの内容をポリシー・フレーム 321 に表示するためのボタンである。Delete ボタン 314 は選択したポリシークラスを削除するためのボタンである。Insert ボタン 315 はテキストフィールド 316 に入力した名称をもつあたらしい空のポリシークラスを生成するためのボタンである。

【0024】ポリシー・フレーム 321 にはポリシー・リスト 322、Show ボタン 323、Delete ボタン 324、Insert ボタン 325、テキストフィールド 326、Show assignment ボタン 327 がふくまれている。ポリシー・リスト 322 には当該のポリシークラスに属するポリシーの名称のリストが表示されるが、マウスを使用してポリシーを選択することができる。Show ボタン 323 は選択したポリシークラスの内容をポリシー・フレーム 331 に表示するためのボタンである。Delete ボタン 324 は選択したポリシーを削除するためのボタンである。Insert ボタン 325 はテキストフィールド 326 に入力した名称をもつあたらしい空のポリシーを生成するためのボタンである。Show assignment ボタンをおすとポリシー適用対象ウィンドウ 341 があたりにひらかれて、ポリシーの適用対象 (policy targets) のリストを表示する。

【0025】規則フレーム 331 には規則リスト 332、S

how ボタン 333、Delete ボタン 334、Insert ボタン 335、テキストフィールド 336 がふくまれている。規則リスト 332 にはポリシー規則名のリストが表示されるが、マウスを使用してポリシー規則を選択することができる。Edit ボタン 333 は選択したポリシー規則の内容をポリシー編集のためのあたなウィンドウをひらいて表示するためのボタンである。Delete ボタン 334 は選択したポリシー規則を削除するためのボタンである。Insert ボタン 335 はテキストフィールド 336 に入力した名称をもつあたらしい空のポリシー規則を生成するためのボタンである。

【0026】ポリシー適用対象ウィンドウ 341 にはポリシー適用対象リスト 342、Install ボタン 343、Remove ボタン 344 がふくまれている。ポリシー適用対象リスト 342 にはポリシーを適用すべき機器やその部分をもつ名称がリストされる。たとえば、ネットワーク機器であればインタフェース名が表示される。さらに、本実施例においてはポリシーエージェント 215、216、217 もポリシー適用対象となるため、これらの名称も表示される。ポリシー適用対象リスト 342 に表示された適用対象を、マウスを使用して選択することができる。複数の適用対象を同時に選択することも可能である。なお、ポリシー適用対象のリストはポリシーエージェント 215、216、217 からポリシーマネージャ 214 を経由してオペレータ・コンソール 211 に定期的に通知されるので、オペレータ・コンソール 211 通知された内容をポリシー適用対象リスト 342 に表示する。

【0027】オペレータ・コンソール 211 には以上のほかにポリシー配布や削除の結果やエラーメッセージのメッセージを表示するためのウィンドウも必要である。

【0028】図 16 および図 4 から図 11 は、オペレータ・コンソール 211 をとおして入力すべきポリシーである。以下これらのポリシーの形式および意味を説明するが、これらのポリシーの意味は図 12 以降の処理手順によって実現されている。なお、以下の記述においてポリシー規則の条件部および動作部はいずれもポリシー変数とその値との組のならばとなっているが、この形式は従来の技術にあげたインターネット・ドラフト draft-ietf-policy-pcim-ext-05.txt に準拠している。

【0029】図 16 にはポリシークラス PolicyDefinition のポリシーの内容が記述されている。PolicyDefinition ポリシー (PolicyDefinition クラスのポリシー) にふくまれるポリシー規則は規則は機器独立なポリシークラスの定義を記述する。PolicyDefinition ポリシーにおいては定義すべきポリシークラスの名称、当該ポリシークラスに属するポリシーにふくまれるポリシー規則の条件部に出現しうる変数名のリスト、動作部に出現しうる変数名のリストが指定される。PolicyDefinition ポリシーの存在により、当該ポリシー規則に不正な変数名が出現していないかどうかを、ポリシーマネージャ 2



12 が検査することが可能になる。

【0030】図16はポリシーを定義するための PolicyDefinition ポリシー 1601 をしめしている。ポリシーヘッダ 1602 には当該のポリシー名 "Policies" とそのポリシークラス PolicyDefinition とが指定されている。ポリシークラス PolicyDefinition はポリシーサーバ 112 にあらかじめくみこまれている。

【0031】ポリシー規則 1611 はポリシークラス QoS のポリシーを定義する。ポリシー規則 1611 の条件部 1612 には「変数 "name" の値が "QoS" であれば」という条件が指定されている。すなわち、条件部 412 においてはポリシー規則 411 において定義すべきポリシークラス名が "QoS" であるということが指定されている。

【0032】また、ポリシー規則 1611 の動作部 1613 にはつぎの各内容が指定されている。ポリシー変数指定 1614 においては、condition\_variables という名称のポリシー変数の値を user\_class' という 1 個の値によって構成されるリストとしている。ポリシー変数指定 1614 はポリシークラス QoS に属するポリシー（以下QoS ポリシーとよぶ）の条件部に指定することができるポリシー変数名を指定している。すなわち、QoS ポリシーの条件部には user\_class' という名称のポリシー変数だけを指定することができる。

【0033】ポリシー変数指定 1615 においては、action\_variables という名称のポリシー変数の値を access' という 1 個の値によって構成されるリストとしている。ポリシー変数指定 1615 は QoS ポリシーの動作部に指定することができるポリシー変数名を指定している。すなわち、QoS ポリシーの動作部には access' という名称のポリシー変数だけを指定することができる。

【0034】図4, 5, 6にはポリシークラス PolicyToCommands のポリシーの内容が記述されている。PolicyToCommands ポリシーにふくまれるポリシー規則は指定されたポリシークラスに属するポリシーからコマンドへの機器依存の変換法を記述する。PolicyToCommands ポリシーにおいては当該ポリシークラスの名称、当該ポリシーに属するポリシー規則から生成されるコマンド列のテンプレートとテンプレートのパラメタ（可変部）をうめるための方法の記述（充填器）、そのコマンド例の生成前に実行すべき初期化处理、ポリシー規則対応の処理の前後に生成すべきコマンド列のテンプレートとその充填器、ポリシー規則対応の処理の前に実行すべき初期化处理が指定される。

【0035】図4は LAN スイッチ 121 用にポリシーを変換できるようにするための SwitchCommands ポリシー 401 の内容を記述している。ポリシーヘッダ 402 には当該のポリシー名 "SwitchCommands" とそのポリシークラス PolicyToCommands とが指定されている。ポリシークラス PolicyToCommands はポリシーサーバ 112 にあ

らかじめくみこまれている。

【0036】ポリシー規則 411 はポリシークラス QoS のポリシーを変換する方法を定義する。ポリシー規則 411 の条件部 412 には「変数 "name" の値が "QoS" であれば」という条件が指定されている。すなわち、条件部 412 においてはポリシー規則 411 において定義すべきポリシークラス名が "QoS" であるということが指定されている。

【0037】また、ポリシー規則 411 の動作部 413 にはつぎの各内容が指定されている。ポリシー変数指定 417 においては、policy\_installation\_epilogue という名称のポリシー変数の値を Table と InterfaceTable[Interface], "ICC", dontCare, ACL, 1 という 5 個の値によって構成されるリストとを要素とする 2 要素のリストとしている。policy\_installation\_epilogue の値は、QoS ポリシー配布時に、その QoS ポリシーに属するすべてのポリシー規則の配布終了後の後処理のためのコマンド列の生成法を記述している。InterfaceTable は QoS ポリシーの配布対象である LAN スイッチのハードウェア・テーブルである。InterfaceTable[Interface] において Interface は QoS ポリシーが適用される機器インタフェースの番号 i をあらわす。したがって、InterfaceTable[Interface] は InterfaceTable の i 番目の要素を意味する。Interface の値はポリシー規則 411 を配布されたエージェントが当該機器にといあわせることによってもとめられる。"ICC" は dontCare, ACL, 1 という値を InterfaceTable に格納する書式をあらわす。すなわち、先頭の 2 バイトはもとのままとし、ACL という局所変数の値（アクセスリスト番号）はつづく 1 バイトに格納し、さらにつづく 1 バイトに 1 を格納する。ここで dontCare は格納すべき値ではなく、値を格納しないことをあらわしている。

【0038】LAN スイッチ 121 の仕様により、InterfaceTable の i 番目の要素は i 番目の機器インタフェースを制御するのに使用される。3 バイトめに格納される値は AccessControlTable の要素番号をあらわし、AccessControlTable の当該要素があらわすアクセス制御または QoS 制御が当該インタフェースに適用される。AccessControlTable は QoS ポリシーの配布対象である LAN スイッチのハードウェア・テーブルである。4 バイトめに格納される値は InterfaceTable の当該要素の有効性をあらわす。すなわち、0 が格納されれば当該要素は無効であり、それ以外の値が格納されれば当該要素は有効である。ポリシー変数指定 417 においては 1 が格納されるため、当該要素を有効にすることを指示している。

【0039】ポリシー変数指定 418 においては、policy\_removal\_prologue という名称のポリシー変数の値を Table と InterfaceTable[Interface], "ICC", dontCare, ACL, 0 という 5 個の値によって構成されるリストとを要素とする 2 要素のリストとしている。policy\_remo



val\_prologue の値は、QoS ポリシー削除時に、その QoS ポリシーに属するすべてのポリシー規則の削除終了後の前処理のためのコマンド列の生成法を記述している。ポリシー変数指定 417 とのちがいは 4 バイトめに格納される値が 1 ではなく 0 である点である。すなわち、ポリシー変数指定 418 においては InterfaceTable の当該要素を無効にすることを指示している。

【0040】ポリシー変数指定 416 においては、rule\_initialization という名称のポリシー変数の値を { ACL = list\_number(ACL', 0, 1023) } という値すなわち ACL という変数名と list\_number(ACL', 0, 1023) という関数およびだしとからなる 1 個の対によって構成される値にしている。ポリシー変数指定 416 の意味は、QoS ポリシーをコマンドに変換する際に、まず ACL という名称の局所変数 (変換のために使用するワーク変数) の値を list\_number(ACL', 0, 1023) という関数およびだしの結果としてえられる値にするという意味である。

【0041】ポリシー変数指定 419 においては、rule\_installation\_commands という名称のポリシー変数の値を Table と AccessControlTable[ACL], "CCIL", 1, (condition(access') == deny' ? 0 : condition(access') == permit' | condition(access') == low\_priority' ? 1 : condition(access') == high\_priority' ? 2), dontCare, condition(user\_class') || 0 という 6 個の値によって構成されるリストとを要素とする 2 要素のリストとしている。rule\_installation\_commands の値は、QoS ポリシー規則配布のためのコマンド列の生成法を記述している。AccessControlTable[ACL] において ACL は局所変数の値をあらわし、AccessControlTable[ACL] はその値によって指示される AccessControlTable の要素をあらわす。

【0042】"CCIL" は前記のリストのそれ以下の要素を AccessControlTable に格納する書式をあらわす。すなわち、まず先頭の 1 バイトには 1 を格納する。つづく 1 バイトには当該 QoS ポリシー規則の条件部にあらわれるポリシー変数 access の値に応じて 0, 1, または 2 を格納する。ここで 2 バイトめの値はつぎのようにして決定される。つづく 2 バイトの値は変更しない。つづく 8 バイトには、もし user\_class というポリシー変数が当該 QoS ポリシー規則の条件部にあらわれていれば、そのポリシー変数の値にポリシー SwitchValue Translation 801 を適用してえられる 8 バイトの値を格納するが、もしそれが条件部にあらわれていなければ 0 を格納する。ポリシー変数 access の値が deny' なら 0, permit' または low\_priority' なら 1, high\_priority' なら 2 とする。

【0043】ポリシー変数指定 420 においては、rule\_removal\_commands という名称のポリシー変数の値を AccessControlTable[ACL], "CC", 0, 0 という 4 個の値によって構成されるリストを唯一の要素とするリストと

している。rule\_removal\_commands の値は、QoS ポリシー規則削除のためのコマンド列の生成法を記述している。AccessControlTable[ACL] において ACL は局所変数の値をあらわし、AccessControlTable[ACL] はその値によって指示される AccessControlTable の要素をあらわす。"CC" は前記のリストのそれ以下の要素を AccessControlTable に格納する書式をあらわす。すなわち、先頭の 1 バイトに 0 を格納し、つづく 1 バイトにも 0 を格納する。

【0044】図 5 はサーバ 131, 132 にポリシーを定義するための ServerCommands ポリシー 501 の内容を記述している。ポリシーヘッダ 502 には当該のポリシー名 "ServerCommands" とそのポリシークラス PolicyToCommands とが指定されている。

【0045】ポリシー規則 511 は QoS ポリシーを変換する方法を定義する。ポリシー規則 511 の条件部 512 には「変数 "name" の値が "QoS" であれば」という条件が指定されている。すなわち、条件部 512 においてはポリシー規則 511 において定義すべきポリシークラス名が "QoS" であるということが指定されている。

【0046】また、ポリシー規則 511 の動作部 513 にはつぎの各内容が指定されている。ポリシー変数指定 516 においては、rule\_installation\_commands という名称のポリシー変数の値を CLI と "allocate-servers %s %s", condition(user\_class'), action(access') == deny' ? 0% : action(access') == permit' | action(access') == low\_priority' ? 20% : action(access') == high\_priority' ? 50%, condition(user\_class') という 3 個の値によって構成されるリストとを要素とする 2 要素のリストとしている。rule\_installation\_commands の値は、QoS ポリシー規則配布のためのコマンド列の生成法を記述している。

【0047】前記のリストの最初の要素 CLI は、当該リストの内容がコマンドライン・インタフェースにおけるコマンドを生成するのに使用されることを表現している。第 2 の要素 "allocate-servers %s %s" は生成すべきコマンドのテンプレートである。すなわち、当該リストからは 2 個のパラメタをもつ allocate-servers コマンドが生成される。パラメタの値は当該リストのつづく 2 個の要素によって決定される。すなわち、第 1 のパラメタとしては当該ポリシー規則の条件部に出現すべきポリシー変数 user\_class の値にポリシー ServerValueTranslation 901 を適用した結果が使用される。また、第 2 のパラメタとしては当該ポリシー規則の動作部に出現するポリシー変数 access の値が deny' ならば 0%, permit' または low\_priority' ならば 20%, high\_priority' ならば 50% という値が使用される。ポリシー変数指定 517 においては、rule\_removal\_commands という名称のポリシー変数の値を CLI と "allocate-servers %s %s", condition(user\_class'), "de

fault” という 3 個の値によって構成されるリストとを要素とする 2 要素のリストとしている。rule\_removal\_commands の値は、QoS ポリシー規則削除のためのコマンド列の生成法を記述している。

【0048】前記の第 1 のリストの最初の要素 CLI は、当該リストの内容がコマンドライン・インタフェースにおけるコマンドを生成するのにつかわれることをあらわしている。前記の第 2 のリストの最初の要素 “allocate-servers %s %s” は生成すべきコマンドのテンプレートである。すなわち、当該リストからは 2 個の

パラメタをもつ allocate-servers コマンドが生成される。パラメタの値は当該リストのつづく 2 個の要素によってきめられる。すなわち、第 1 のパラメタとしては当該ポリシー規則の条件部にあらわれるべきポリシー変数 user\_class の値にポリシー ServerValueTranslation 901 を適用した結果がつかわれる。また、第 2 のパラメタとしては “default” という文字列がつかわれる。

【0049】サーバの仕様により、allocate-servers コマンドはサーバ 131, 132 の CPU 時間などの資源をどれだけ特定のユーザにわりあてるかを指定する。すなわち、 “allocate-servers User 50%” というコマンドは、User という名称のユーザに 50% の資源をあたえることを指示する。

【0050】図 6 はストレージ 150, 151, 152 にポリシーを定義するための StorageCommands ポリシー 601 の内容を記述している。ポリシーヘッダ 602 には当該のポリシー名 “StorageCommands” とそのポリシークラス PolicyToCommands とが指定されている。

【0051】ポリシー規則 611 は QoS ポリシーを変換する方法を定義する。ポリシー規則 611 の条件部 612 には「変数 “name” の値が “QoS” であれば」という条件が指定されている。すなわち、条件部 612 においてはポリシー規則 611 において定義すべきポリシークラス名が “QoS” であるということが指定されている。

【0052】また、ポリシー規則 611 の動作部 613 にはつぎの各内容が指定されている。ポリシー変数指定 616 においては、rule\_installation\_commands という名称のポリシー変数の値を MIB, set(storage\_mib.caching.unit, condition(user\_class')), set(storage\_mib.caching.strategy, action(access') == deny' | action(access') == permit' | action(access') == low\_priority' ? nocache' : action(access') == high\_priority' ? cache') という 3 個の値によって構成されるリストとしている。rule\_installation\_commands の値は、QoS ポリシー規則配布のためのコマンド列の生成法を記述している。

【0053】前記のリストの最初の要素 MIB は、当該リストの内容が SNMP インタフェースにおけるコマンドを生成するのにつかわれることをあらわしている。第 2

および第 3 の要素は生成すべきコマンドのテンプレートである。すなわち、当該リストからは 2 個の set コマンドが生成され、MIB (Management Information Base) の 2 個の要素の値が設定される。これら 2 個の set コマンドによってデータベース・ファイルをキャッシュすること、またはしないことが指示される。すなわち、第 1 の set コマンドによってキャッシングの対象となるファイルが指示され、第 2 の set コマンドによってキャッシングをおこなうこと (cache') またはおこなわないこと (nocache') が指示される。第 1 の set コマンドのパラメタとしては当該ポリシー規則の条件部にあらわれるべきポリシー変数 user\_class の値にポリシー StorageValueTranslation 1001 を適用した結果がつかわれる。また、第 2 の set コマンドのパラメタとしては当該ポリシー規則の動作部にあらわれるポリシー変数 access の値が deny', permit' または low\_priority' ならば nocache', high\_priority' ならば cache' という値が使用される。

【0054】ポリシー変数指定 617 においては、rule\_removal\_commands という名称のポリシー変数の値を MIB, set(storage\_mib.caching.unit, condition(user\_class')), set(storage\_mib.caching.strategy, nocache') という 3 個の値によって構成されるリストとしている。rule\_removal\_commands の値は、QoS ポリシー規則削除のためのコマンド列の生成法を記述している。前記のリストの最初の要素 MIB は、当該リストの内容が SNMP インタフェースにおけるコマンドを生成するのにつかわれることをあらわしている。第 2 および第 3 の要素は生成すべきコマンドのテンプレートである。すなわち、当該リストからは 2 個の set コマンドが生成され、MIB (Management Information Base) の 2 個の要素の値が設定される。これら 2 個の set コマンドによってデータベース・ファイルをキャッシュしないことが指示される。

【0055】図 7 にはポリシークラス PolicyVariable Definition のポリシーの内容が記述されている。PolicyVariableDefinition ポリシーにふくまれるポリシー規則は、ポリシー規則の条件部にあらわれる、指定された名称をもつポリシー変数がとりうる値の型のリストとが指定される。

【0056】図 7 はポリシー変数がとりうる値の型を指定するための VariableDefinition ポリシー 701 の内容を記述している。ポリシーヘッダ 702 には当該のポリシー名 “VariableDefinition” とそのポリシークラス PolicyVariableDefinition とが指定されている。

【0057】ポリシー規則 711 は user\_class というポリシー変数がとりうる値の型を指定する。ポリシー規則 711 の条件部 712 には「変数 “name” の値が “user\_class” であれば」という条件が指定されている。すなわち、条件部 712 においてはポリシー規則 711 におい

て指定するべきポリシー変数名が "user\_class" であるということが指定されている。

【0058】また、ポリシー規則 711 の動作部 713 においては、ポリシー変数 value\_type の値を user\_class' という値だけからなるリストとしている。これは、user\_class というポリシー変数を使用するポリシー規則において、user\_class の値としてはポリシー SwitchValueTranslation 801, ServerValueTranslation 901, StorageValueTranslation 1001 において定義される user\_class 型だけがゆるされることをあらわしている。

【0059】図 8, 9, 10 にはポリシークラス PolicyValueTranslation のポリシーの内容が記述されている。PolicyValueTranslation ポリシーにふくまれるポリシー規則は、ポリシー規則の条件部にあらわれる、指定された名称をもつポリシー変数の値から、ポリシー規則を変換してえられるコマンドにおいて使用される値への機器依存の変換法を記述する。

【0060】図 8 はポリシー変数にとる値をポリシー配布対象の LAN スイッチにあわせて変換するための SwitchValueTranslation ポリシー 801 の内容を記述している。ポリシーヘッダ 802 には当該のポリシー名 "SwitchValueTranslation" とそのポリシークラス PolicyValueTranslation とが指定されている。

【0061】ポリシー規則 811 は user\_class 型の値の変換法を定義する。ポリシー規則 811 の条件部 812 には「変数 "name" の値が "user\_class" であれば」という条件が指定されている。すなわち、条件部 812 においてはポリシー規則 811 において変換法を定義するべき値の型名が "user\_class" であることを指定している。

【0062】また、ポリシー規則 811 の動作部 813 においては、ポリシー変数 result の値を (value == Sales' ? 10.1.0.0/255.255.0.0 : value == Development' ? 10.2.0.0/255.255.0.0) という 1 個の値にすることが指定されている。これは、変換するべき値が Sales' であれば変換後の値を IP アドレス 10.1.0.0 とマスク 255.255.0.0 (すなわち販売部門の IP アドレスとマスク) をつなげた 8 バイトの値とし、変換するべき値が Development' であれば変換後の値を IP アドレス 10.2.0.0 とマスク 255.255.0.0 (すなわち開発部門の IP アドレスとマスク) をつなげた 8 バイトの値とすることをあらわしている。

【0063】図 9 はポリシー変数にとる値をポリシー配布対象のサーバにあわせて変換するための ServerValueTranslation ポリシー 901 の内容を記述している。ポリシーヘッダ 902 には当該のポリシー名 "ServerValueTranslation" とそのポリシークラス PolicyValueTranslation とが指定されている。

【0064】ポリシー規則 911 は user\_class 型の値の変換法を定義する。ポリシー規則 911 の条件部 912

には「変数 "name" の値が "user\_class" であれば」という条件が指定されている。すなわち、条件部 912 においてはポリシー規則 911 において変換法を定義するべき値の型名が "user\_class" であることを指定している。

【0065】また、ポリシー規則 911 の動作部 913 においては、ポリシー変数 result の値を (value == Sales' ? userid(Mike, Cindy, Dick) : value == Development' ? userid(Nancy, Bob, Bill)) という 1 個の値にすることが指定されている。これは、変換するべき値が Sales' であれば変換後の値を userid(Mike, Cindy, Dick) (すなわち販売部門に所属するユーザのユーザ ID) とし、変換するべき値が Development' であれば変換後の値を userid(Nancy, Bob, Bill) (すなわち開発部門に所属するユーザのユーザ ID) とすることをあらわしている。

【0066】図 10 はポリシー変数にとる値をポリシー配布対象のストレージにあわせて変換するための StorageValueTranslation ポリシー 1001 の内容を記述している。ポリシーヘッダ 1002 には当該のポリシー名 "StorageValueTranslation" とそのポリシークラス PolicyValueTranslation とが指定されている。

【0067】ポリシー規則 1011 は user\_class 型の値の変換法を定義する。ポリシー規則 1011 の条件部 1012 には「変数 "name" の値が "user\_class" であれば」という条件が指定されている。すなわち、条件部 1012 においてはポリシー規則 1011 において変換法を定義するべき値の型名が "user\_class" であることを指定している。

【0068】また、ポリシー規則 1011 の動作部 1013 においては、ポリシー変数 result の値を (value == Sales' ? SalesDB : value == Development' ? ProductDB) という 1 個の値にすることが指定されている。これは、変換するべき値が Sales' であれば変換後の値を SalesDB (すなわち販売部門が使用するデータベース・ファイル名またはディスク・ボリューム名) とし、変換するべき値が Development' であれば変換後の値を ProductDB (すなわち開発部門が使用するデータベース・ファイル名またはディスク・ボリューム名) とすることをあらわしている。

【0069】図 11 は、図 16 および図 4 から図 10 のポリシー定義によって定義された QoS ポリシーの定義例をしめす。すなわち、LAN スイッチ 121, サーバ 131, 132, ストレージ 150, 151, 152 に適用することによって、X 社の販売部門にシステムを優先使用させるための QoS ポリシー 1101 を定義する。ポリシーヘッダ 1102 には当該のポリシー名 "PrioritizeMarketing" とそのポリシークラス QoS とが指定されている。

【0070】ポリシー規則 1111 は時刻 9:00 から 10:00 までのあいだ販売部門のトランザクションを開発部

門のトランザクションより優先することを指示するポリシー規則である。すなわち、ポリシー規則 1111 の条件部 1112 にはポリシー規則 1111 が適用されるべき時刻が 9:00 から 10:00 であることを指定する第 1 の条件と、ポリシー規則 1111 が適用されるべきユーザのクラスが販売部門 (Marketing') であることが指定されている。また、ポリシー規則 1111 の動作部 1113 にはトランザクションを優先すること (high\_priority') が指定されている。以下、図 1 2, 1 3, 1 4 を使用してポリシーマネージャ 212 の構成とポリシーマネージャ 212 における処理手順とを説明する。

【0071】図 1 2 はポリシーマネージャ 212 の構成図である。ポリシーマネージャ 212 はオペレータ・コンソール 211 からの要求を上位受信部 1211 において受信し、当該要求をマネージャ制御部 1212 にわたし、マネージャ制御部 1212 からえられる結果を結果を上位受信部 1211 をととしてオペレータ・コンソール 211 にかえす。オペレータ・コンソール 211 からの要求種別はつぎのうちのいずれかである。

【0072】create(name, body), delete(name), get(name), install(name, target), remove(name, target)。ここで name はエンティティすなわちポリシークラス、ポリシーまたはポリシー規則の名称であり、body は生成するエンティティの本体であり、target はポリシー適用対象の名称である。エンティティおよびポリシー適用対象は複数個わたされるばあいもある。上記の要求種別のなかには更新のための種別が存在しないが、更新は create により既存の定義をうわがきするか、または delete によって既存の定義を削除してから create によって定義し直すことによって実現することができる。name が ":" をふくまないときはポリシークラス名をあらわし、1 個の ":" をふくむときはポリシークラス名とポリシー名とを ":" をはさんでつないだものとみなす。2 個の ":" をふくむときはポリシークラス名とポリシー名とポリシー規則名とをそれぞれ ":" をはさんでつないだものとみなす。すなわち、上記のインタフェースによって、ポリシークラス、ポリシー、ポリシー規則のいずれの種類のエンティティも生成・削除することができる。

【0073】これらの要求の処理方法は後述するが、ここでその機能だけを説明する。create 要求は指定された名称 (name) をもつエンティティを生成し、ポリシー・リポジトリ 213 に登録するとともに、必要に応じてポリシーマネージャやエージェントの内部データベースに格納することをもとめる。delete 要求は指定された名称をもつエンティティをポリシー・リポジトリ 213 およびポリシーマネージャやエージェントの内部データベースから削除することをもとめる。get 要求は指定された名称をもつエンティティの内容をポリシー・リポジトリ 213 からとりだすことをもとめる。install 要求は指

定された名称をもつポリシーを指定されたポリシー適用対象に適用することをもとめる。また remove 要求は指定された名称をもつポリシーを指定されたポリシー適用対象から削除することをもとめる。

【0074】Install 要求のなかにはサーバポリシーに関する要求とそうでない要求とがあるが、前者のばあいにはサーバポリシー・コンパイラ 1213 を使用して情報を内部形式に変換してからポリシーマネージャ 212 の内部データベースであるサーバポリシー・データベース 1214 に登録する。また、サーバポリシーに関する要求でないばあいには、ポリシーをサーバポリシー・データベース 1214 の情報にもとづいて検査してから中位送信部 1216 をととしてエージェントに送付する。

【0075】つぎに、図 1 3 を使用してマネージャ制御部 1212 におけるオペレータ・コンソール 211 からの要求の処理方法を説明する。図 1 3 (a) には create 要求の処理方法を記述している。create 要求が上位受信部 1211 からつたえられると、まずステップ 1311 において当該のエンティティをポリシー・リポジトリ 213 に登録する。すなわち、当該のエンティティ名をキーとし、エンティティ本体を値として登録する。ポリシークラスおよびポリシーの登録においてはエンティティ本体は空だが、ポリシー規則の登録においてはポリシー規則本体をとともなう。そして、図 1 3 (a) の処理を終了する。

【0076】図 1 3 (b) には delete 要求の処理方法を記述している。delete 要求が上位受信部 1211 からつたえられると、まずステップ 1321 において当該のエンティティを、当該のエンティティ名をキーとして、ポリシー・リポジトリ 213 から削除する。そして、図 1 3 (b) の処理を終了する。

【0077】図 1 3 (c) には get 要求の処理方法を記述している。get 要求が上位受信部 1211 からつたえられると、まずステップ 1331 において当該のエンティティを、当該のエンティティ名をキーとして、ポリシー・リポジトリ 213 から検索する。ステップ 1332 において、えられたエンティティ本体をよびだしもにかえて、図 1 3 (c) の処理を終了する。

【0078】図 1 3 (d) には install 要求および remove 要求の処理方法を記述している。install 要求または remove 要求が上位受信部 1211 からつたえられると、まずステップ 1341 において当該のエンティティを、当該のエンティティ名をキーとして、ポリシー・リポジトリ 213 からさがしだす。つづいて、ステップ 1342 において当該のエンティティがサーバポリシーであるかどうかを判定する。ここでサーバポリシーとはサーバの動作に影響をあたえるポリシーであり、PolicyDefinition ポリシーおよび PolicyVariableDefinition ポリシーがそれにあたる。当該のエンティティがこれらのポリシーであるばあいにはステップ 1343 にすすみ、そう

でないばあいにはステップ 1347 にすすむ。

【0079】ステップ 1343 においては要求が install であるか remove であるかを判定する。install のばあいはステップ 1344 にすすみ、remove のばあいはステップ 1346 にすすむ。ステップ 1344 においては、当該のポリシー（エンティティ）をサーバポリシー・コンパイラ 1213 をよびだして、マネージャ制御部 1212 において使用するのに適した形式に変換する。そして、ステップ 1345 においてその結果すなわち 1 対または 2 対のキーと値とをサーバポリシー・データベース1214 に格納して、図 13 (d) の処理を終了する。サーバポリシー・データベース 1214 は文字列のキーと値の対を登録することができるデータベースであり、たとえば GNU データベース・マネージャ (GDBM) のようなハッシュ表によって実現することができる。ステップ 1346 においては当該のポリシーをサーバポリシー・データベース 1214 から削除して、図 13 (d) の処理を終了する。

【0080】ステップ 1347 においては、ポリシーを配布するばあいにかぎり、ポリシー検査部 1215 をよびだして当該ポリシーの妥当性を判定する。ポリシー・リポジトリ 213 の検索によってえられたエンティティ本体を要求種別 (install または remove)、エンティティ名とともに下位送信部 1215 をとおして、要求において指定されているポリシー適用対象を管理しているエージェントに送信して、図 13 (d) の処理を終了する。

【0081】つぎに、図 14 を使用してサーバポリシー・コンパイラ 1213 の処理手順を説明する。サーバポリシー・コンパイラ 1213 の実行が開始されると、まずステップ 1411 において入力されたポリシーのポリシークラスを判定する。ポリシークラスが PolicyDefinition であればステップ 1412 にすすみ、PolicyVariableDefinition であればステップ 1413 にすすむ。

【0082】ステップ 1412 においては、入力されたポリシーの条件部において値が定義されるポリシー変数名 (ポリシー変数 name の値) に ":type" という文字列を付加して、サーバポリシー・データベース 1214 に値の型のリストを登録する際に使用するキーを生成する。たとえば、ポリシー変数名が "user\_class" であれば "user\_class:type" というキーを生成する。また、動作部において値が定義される値の型のリスト (ポリシー変数 value\_type の値) をとりだして、登録データとする。

【0083】ステップ 1413 においては、入力されたポリシーの条件部において値が定義されるポリシークラス名 (ポリシー変数 name の値) に ":condition" という文字列を付加して、サーバポリシー・データベース 1214 に条件部ポリシー変数リストを登録する際に使用するキーを生成する。たとえば、ポリシー変数名が "QoS" であれば "QoS:condition" というキーを生成する。また、動作部において値が定義されるポリシー変数 condition\_variable の値をとりだして、条件部ポリシー変数

リストのための登録データとする。

【0084】ステップ 1414 においては、入力されたポリシーの条件部において値が定義されるポリシークラス名 (ポリシー変数 name の値) に ":action" という文字列を付加して、サーバポリシー・データベース 1214 に動作部ポリシー変数リストを登録する際に使用するキーを生成する。たとえば、ポリシー変数名が "QoS" であれば "QoS:action" というキーを生成する。また、動作部において値が定義されるポリシー変数 action\_variable の値をとりだして、動作部ポリシー変数リストのための登録データとする。

【0085】つぎに、図 15 を使用してポリシー検査部 1215 の処理手順を説明する。ポリシー検査部 1215 の実行が開始されると、まずステップ 1512 において、入力されたポリシーの条件部において値が定義されるすべてのポリシー変数名を検査する。すなわち、当該ポリシーのポリシークラス名に ":condition" を付加した文字列をキーとしてサーバポリシー・データベース 1214 からポリシー変数名のリストをとりだし、当該ポリシー変数名がそのリストにふくまれるかどうかをテストする。ふくまれないものがあればエラーメッセージをオペレータ・コンソール211にかえす。

【0086】つぎにステップ 1512 において、入力されたポリシーの条件部において値が定義されるすべての値に対して、その型が、対応する変数名に対して許容されたものであるかどうかを検査する。すなわち、当該ポリシー変数名に ":type" を付加した文字列をキーとしてサーバポリシー・データベース 1214 から値の型のリストをとりだし、当該の値の型がそのリストにふくまれるかどうかをテストする。ふくまれないものがあればエラーメッセージをオペレータ・コンソール 211にかえす。

【0087】最後にステップ 1513 において、入力されたポリシーの動作部にふくまれるすべてのポリシー変数名を検査する。すなわち、当該ポリシーのポリシークラス名に ":action" を付加した文字列をキーとしてサーバポリシー・データベース 1214 からポリシー変数名のリストをとりだし、当該ポリシー変数名がそのリストにふくまれるかどうかをテストする。ふくまれないものがあればエラーメッセージをオペレータ・コンソール 211にかえす。

【0088】図 17 はポリシーエージェント 215、216、217 の構成図である。ネットワーク・エージェント 215、サーバ・エージェント 216、ストレージ・エージェント217 の構成はひとしい。ポリシーエージェント 215、216、217 はポリシーマネージャ 212 からの要求を中位受信部 1711 において受信し、当該要求をエージェント制御部 1712 にわたし、エージェント制御部 1712 からえられる結果を結果を中位受信部 1711 をとおしてポリシーマネージャ 212 にかえす。ポリシーマネージャ 212 からの要求はつぎのうちのいずれかである。

【0089】install(name, target), remove(name, target). ここで name はエンティティすなわちポリシークラス、ポリシーまたはポリシー規則の名称であり、target はポリシー適用対象の名称である。エンティティおよびポリシー適用対象は複数個わたされるばあいもある。

【0090】Install 要求のなかにはエージェント・ポリシーに関する要求とそうでない要求とがあるが、前者のばあいにはエージェント・ポリシー・コンパイラ 1713 を使用して情報を内部形式に変換してからポリシーエー  
10 ジェント 215、216、217 の内部データベースであるエージェント・ポリシー・データベース 1714 に登録する。また、エージェント・ポリシーに関する要求でないばあいには、ポリシー変換部 1715 においてポリシーを  
エージェント・ポリシー・データベース 1714 の情報にもとづいて変換してから機器設定部 1718 をとおしてエー  
ジェントに送付する。ポリシー変換部 1715 はポリシー全体を変換するが、生成するべきコマンド列ごとにコ  
マンド変換部 1716 がよびだされる。また、コマンド変  
20 換部 1716 は評価するべき式をインタプリタ 1717 にわ  
たしてその値をもとめる。

【0091】つぎに図 18 を使用してネットワーク・エー  
ジェント 215、サーバ・エージェント 216、ストレ  
ジ・エージェント 217 における要求の処理方法を説明  
する。エージェント 215、216、217 がポリシーマネ  
ジャ 212 からうけとる要求として install 要求と rem  
ove 要求とがある。install 要求または remove 要求が中  
位受信部 1711 からつたえられると、まずステップ 181  
1 において当該のエンティティがエージェント・ポリ  
シーであるかどうかを判定する。ここでエージェント・ポリ  
30 シーとはエージェントの動作に影響をあたえるポリ  
シーであり、PolicyToCommands ポリシーおよび PolicyVa  
lueTranslation ポリシーがそれにあたる。当該のエン  
ティティがこれらのポリシーであるばあいにはステップ  
1812 にすすみ、そうでないばあいにはステップ 1816  
にすすむ。

【0092】ステップ 1812 においては要求が install  
であるか remove であるかを判定する。install のば  
あいはステップ 1813 にすすみ、remove のばあいはス  
テップ 1815 にすすむ。ステップ 1813 においては、当  
40 該のポリシー（エンティティ）をエージェント・ポリ  
シー・コンパイラ 1713 を使用してエージェント制御部 17  
12 において使用するのに適した形式に変換する。そし  
て、ステップ 1814 においてその結果すなわち 1 対ま  
たは複数対のキーと値とをエージェント・ポリシー・デ  
ータベース 1714 に格納して、図 18 の処理を終了す  
る。ステップ 1815 においては当該のポリシーをエー  
ジェント・ポリシー・データベース 1714 から削除して、  
図 18 の処理を終了する。

【0093】ステップ 1816 においては、ポリシーを配 50

布するばあいにかぎり、ポリシー変換部 1715 をよびだ  
して当該ポリシーをコマンドに変換する。変換結果のコ  
マンドを機器設定部 1718 をとおして、要求において指  
定されているポリシー適用対象に設定して、図 18 の処  
理を終了する。

【0094】つぎに、図 19 を使用してエージェント・  
ポリシー・コンパイラ 1713 の処理手順を説明する。エー  
ジェント・ポリシー・コンパイラ 1713 の実行が開始  
されると、まずステップ 1911 において入力されたポリ  
シーのポリシークラスを判定する。ポリシークラスが P  
olicyToCommands であればステップ 1912 にすすみ、Po  
lityValueTranslation であればステップ 1921 にすす  
む。

【0095】ステップ 1912 においては、入力されたポリ  
シーの条件部にふくまれるポリシークラス名（ポリシ  
ー変数 name の値）に ":policy\_init" という文字列を  
付加して、エージェント・ポリシー・データベース 171  
4 にポリシー変換開始時の初期化リストを登録する際に  
使用するキーを生成する。たとえば、ポリシークラス名  
が "QoS" であれば "QoS:policy\_init" というキーを生  
成する。また、動作部にふくまれるポリシー変数 polic  
y\_initialization に対応する値（局所変数とその初期  
値との対のリスト）をとりだして、登録データとする。

【0096】ステップ 1913 においては、入力されたポリ  
シーによって変換法が定義されるポリシーがふくむポリ  
シー規則の配布時および削除時に生成する初期化およ  
び終期化のための 2 組のコマンド・テンプレートとテ  
ンプレートの充填器とをエージェント・ポリシー・デー  
タベース 1214 に登録するために準備する。すなわち、  
30 第 1 に当該ポリシーの条件部にふくまれるポリシーク  
ラス名（ポリシー変数 name の値）に ":installation"  
という文字列を付加して、エージェント・ポリシー・デ  
ータベース 1714 にポリシー規則配布コマンドリストを  
登録する際に使用するキーを生成する。たとえば、ポリ  
シー変数名が "QoS" であれば "QoS:installation" と  
いうキーを生成する。また、動作部にふくまれるポリ  
シー変数 rule\_installation\_commands の値をとりだし  
て、ポリシー規則配布コマンドリストの登録データとす  
る。

【0097】第 2 に当該ポリシーの条件部にふくまれ  
るポリシークラス名（ポリシー変数 name の値）に ":re  
moval" という文字列を付加して、エージェント・ポリ  
シー・データベース 1714 にポリシー規則削除コマンド  
リストを登録する際に使用するキーを生成する。たと  
えば、ポリシー変数名が "QoS" であれば "QoS:removal"  
というキーを生成する。また、動作部にふくまれるポリ  
シー変数 rule\_removal\_commands の値をとりだして、  
ポリシー規則削除コマンドリストの登録データとする。

【0098】ステップ 1914 においては、入力されたポリ  
シーによって変換法が定義されるポリシーの配布時お



および削除時に生成する初期化および終期化のための 4 組のコマンド・テンプレートとテンプレートの充填器とをエージェント・ポリシー・データベース 1214 に登録するために準備する。すなわち、第 1 に当該ポリシーの条件部にふくまれるポリシークラス名 (ポリシー変数 name の値) に ":installation\_prologue" という文字列を付加して、エージェント・ポリシー・データベース 1714 にポリシー配布時初期化コマンドリストを登録する際に使用するキーを生成する。また、動作部にふくまれるポリシー変数 policy\_installation\_prologue の値をとりだして、ポリシー配布時初期化コマンドリストの登録データとする。ただし、ポリシー変数 policy\_installation\_prologue が動作部にふくまれないときは、登録データを空にする。

【0099】第 2 に当該ポリシーの条件部にふくまれるポリシークラス名 (ポリシー変数 name の値) に ":installation\_epilogue" という文字列を付加して、エージェント・ポリシー・データベース 1714 にポリシー配布時終期化コマンドリストを登録する際に使用するキーを生成する。また、動作部にふくまれるポリシー変数 policy\_installation\_epilogue の値をとりだして、ポリシー配布時終期化コマンドリストの登録データとする。ただし、ポリシー変数 policy\_installation\_epilogue が動作部にふくまれないときは、登録データを空にする。

【0100】第 3 に当該ポリシーの条件部にふくまれるポリシークラス名 (ポリシー変数 name の値) に ":removal\_prologue" という文字列を付加して、エージェント・ポリシー・データベース 1714 にポリシー削除時初期化コマンドリストを登録する際に使用するキーを生成する。また、動作部にふくまれるポリシー変数 policy\_removal\_prologue の値をとりだして、ポリシー削除時初期化コマンドリストの登録データとする。ただし、ポリシー変数 policy\_removal\_prologue が動作部にふくまれないときは、登録データを空にする。

【0101】第 4 に当該ポリシーの条件部にふくまれるポリシークラス名 (ポリシー変数 name の値) に ":removal\_epilogue" という文字列を付加して、エージェント・ポリシー・データベース 1214 にポリシー削除時終期化コマンドリストを登録する際に使用するキーを生成する。また、動作部にふくまれるポリシー変数 policy\_removal\_epilogue の値をとりだして、ポリシー削除時終期化コマンドリストの登録データとする。ただし、ポリシー変数 policy\_removal\_epilogue が動作部にふくまれないときは、登録データを空にする。

【0102】ステップ 1920 においては、入力されたポリシーの条件部にふくまれるポリシークラス名 (ポリシー変数 name の値) に ":rule\_init" という文字列を付加して、エージェント・ポリシー・データベース 1714 にポリシー規則変換時の初期化リストを登録する際に使

用するキーを生成する。たとえば、ポリシークラス名が "QoS" であれば "QoS:rule\_init" というキーを生成する。また、動作部にふくまれるポリシー変数 rule\_initialization に対応する値 (局所変数とその値との対のリスト) をとりだして、登録データとする。

【0103】ステップ 1921 においては、入力されたポリシーの条件部にふくまれるポリシーの値の型名 (ポリシー変数 name の値) に ":translation" という文字列を付加して、エージェント・ポリシー・データベース 1714 に値の変換法記述のリストを登録する際に使用するキーを生成する。たとえば、ポリシーの値の型名が user\_class であれば "user\_class:translation" というキーを生成する。また、動作部にふくまれるポリシー変数 result に対応する値 (値の変換法記述のリスト) をとりだして、登録データとする。

【0104】つぎに、図 20 を使用してポリシー変換器 1215 の処理手順を説明する。ポリシー変換器 1215 の実行が開始されると、まずステップ 2011 において局所変数の初期化をおこなう。局所変数のリストはエージェント・ポリシー・データベース 1214 に登録されているので、入力されたポリシーのポリシークラス名に ":policy\_init" という文字列を付加したものをキーとしてエージェント・ポリシー・データベース 1214 をひく。えられたリストは変数名とその初期値との対を要素とするリストであるから、インタプリタ 1717 を使用して初期値を評価し、その結果を当該変数名に束縛する。たとえば、ACL = list\_number(ACL', 0, 1023) という対のばあいには list\_number(ACL', 0, 1023) という式をインタプリタ 1717 によって評価し、その結果としてえられる整数値を変数 ACL の初期値とする。

【0105】つづいて、ステップ 2012 においてポリシー初期化コマンド列を生成する。ポリシーの配布が指示されているとき (install 要求のとき) は当該ポリシーのポリシークラス名に ":installation\_prologue" という文字列を付加したものをキーとしてエージェント・ポリシー・データベース 1214 をひくことによりポリシー配布時初期化コマンドリストをもとめ、ポリシーの削除が指示されているとき (remove 要求のとき) は当該ポリシーのポリシークラス名に ":removal\_prologue" という文字列を付加したものをキーとしてエージェント・ポリシー・データベース 1214 をひくことによりポリシー削除時初期化コマンドリストをもとめる。コマンド変換器 1716 をよびだして、もとめたコマンドリストからコマンド列を生成する。

【0106】つづいて、ステップ 2013 において当該ポリシーがふくむ各ポリシー規則に対応するコマンド列をもとめる。すなわち、ポリシー規則ごとにステップ 2014 を実行する。ステップ 2014 においては、ポリシーの配布が指示されているとき (install 要求のとき) は当該ポリシーのポリシークラス名に ":installation" とい



う文字列を付加したものをキーとしてエージェント・ポリシー・データベース 1214 をひくことによりポリシー規則配布コマンドリストをもとめ、ポリシーの削除が指示されているとき (remove 要求のとき) は当該ポリシーのポリシークラス名に ":removal" という文字列を付加したものをキーとしてエージェント・ポリシー・データベース 1214 をひくことによりポリシー規則削除コマンドリストをもとめる。コマンド変換器 1716 をよびだして、もとのコマンドリストからコマンド列を生成する。

【0107】最後に、ステップ 2015 においてポリシー終期化コマンド列を生成する。ポリシーの配布が指示されているとき (install 要求のとき) は当該ポリシーのポリシークラス名に ":installation\_epilogue" という文字列を付加したものをキーとしてエージェント・ポリシー・データベース 1214 をひくことによりポリシー配布時終期化コマンドリストをもとめ、ポリシーの削除が指示されているとき (remove 要求のとき) は当該ポリシーのポリシークラス名に ":removal\_epilogue" という文字列を付加したものをキーとしてエージェント・

ポリシー・データベース 1214 をひくことによりポリシー削除時終期化コマンドリストをもとめる。コマンド変換器 1716 をよびだして、もとのコマンドリストからコマンド列を生成する。

【0108】上記の各ステップにおいて生成されたコマンド列を生成順につなぎあわせたものが、機器に設定すべきコマンド列である。

【0109】つぎに図 21 を使用してコマンド変換器 1716 の処理手順を説明する。コマンド変換器 1716 の実行が開始されると、まずステップ 2111 において、入力されたコマンドリストがふくむすべてのコマンド・テンプレートについて、ステップ 2112 から 2114 の各ステップを実行する。コマンドリストは、コマンド・テンプレートとそれに対応するパラメタをうめるための充填器のリストとからなる対を要素とするリストである。ステップ 2112 においてはテンプレートのコピーをつくる。ステップ 2113 においてはコピーされたテンプレートがふくむパラメタを順にうめていく。すなわち、テンプレートがふくむすべてのパラメタについてステップ 2114 を実行する。ステップ 2114 においては、コマンドリストがふくむテンプレート充填器をインタプリタ 1717 をよびだして評価し、その結果えられる値によってパラメタをうめる。

【0110】パラメタのうめかたはコマンドの種類によってことなる。第 1 に、生成したコマンドがハードウェア・テーブルにかきこまれるばあい、すなわちコマンドリストの最初の要素がハードウェア・テーブルを意味しているときには、書式にしたがってテーブルをうめていく。たとえば、ポリシー変数と値の対 417 によって指定されたコマンドリストにおいては、その最初の要素

InterfaceTable[Interface] がハードウェア・テーブルを意味しているため、ハードウェア・テーブルのためのコマンド生成がおこなわれる。まず書式 "ICC" の最初の要素 "I" と最初の充填器 dontCare とをとりだし、dontCare を評価する。dontCare は値を格納しないことを意味するので、そのとおりにする。つぎに第 2 の要素 "C" と第 2 の充填器 ACL とをとりだし、ACL をインタプリタ 1717 によって評価してその値をもとめる。ACL は局所変数をあらわすので、その値がもとめられ、それがテーブルの 3 バイトめに格納される。最後に第 3 の要素 "C" と第 3 の充填器 1 とをとりだし、1 をインタプリタ 1717 によって評価する。その値すなわち 1 がテーブルの 4 バイトめに格納される。

【0111】第 2 に、生成すべきコマンドがコマンドライン・インタフェースのコマンドであるとき、すなわちコマンドリストの最初の要素が "CLI" であるときは、1 行ずつコマンドを生成していく。たとえば、ポリシー変数と値の対 517 によって指定されたコマンドリストにおいては 1 行だけのコマンドを生成する。すなわち、まず condition(user\_class') をインタプリタ 1717 によって評価し、"userid(Mike, Cindy, Dick)" という文字列を結果としてえる。つぎに "default" をインタプリタ 1717 によって評価するが、"default" のままである。これらをつかってコマンド・テンプレート "allocate-servers %s %s" のパラメタ "%s" をうめると、"allocate-servers userid(Mike, Cindy, Dick) default" というコマンドがえられる。

【0112】第 3 に、生成すべきコマンドが SNMP による MIB へのかきこみのコマンドであるとき、すなわちコマンドリストの最初の要素が "SNMP" であるときは、set コマンドを 1 個ずつ生成していく。たとえば、ポリシー変数と値の対 617 によって指定されたコマンドリストにおいては、つぎのような 2 個のコマンドを生成する。まず condition(user\_class') をインタプリタ 1717 によって評価し、"SalesDB" という結果をえる。したがって、set(storage\_mib.caching.unit, "SalesDB") というコマンドが生成される。つぎに、"nocache" という文字列をインタプリタ 1717 によって評価するが、そのままの値がかえされるので、set(storage\_mib.caching.strategy, "nocache") というコマンドが生成される。ただし、ここで "SalesDB" および "nocache" という文字列は機器設定部 1718 においてエンコードされてから送信される。

【0113】つぎに図 22 を使用してインタプリタ 1717 の処理手順を説明する。インタプリタ 1717 の実行が開始されると、まずステップ 1911 において解釈すべき式の種類を判定する。種類が演算子をふくむ式であるときはステップ 1912 にすすみ、定数であるときはステップ 1921 にすすみ、局所変数であるときにはステップ 1931 にすすみ、関数およびだしであるときはステップ 1

941 にすすむ。

【0114】ステップ 1912 においては当該の式の各演算項を、インタプリタ 1717 を再帰呼びだしして評価し、結果をえる。ただし、当該の式が条件式すなわち  $x || y$  あるいは  $x ? y : z$  のような形式であるときには、 $x$  だけを評価する。

【0115】つづいてステップ 1913 において、えられた結果を当該の式の演算子にあてて評価する。たとえば、当該の式が  $x == y$  であれば  $x$  と  $y$  の値がひとしきに 1、そうでないときに 0 という値をかえす。また、当該の式が  $x ? y : z$  であれば  $x$  の値が 1 のときにはインタプリタ 1717 を再帰呼びだしして  $y$  を評価してその値をかえす。また、 $x$  の値が 0 のときにはインタプリタ 1717 を再帰呼びだしして  $z$  を評価してその値をかえす。さらに、当該の式が  $condition(x) || y$  または  $action(x) || y$  であるときには、条件部または動作部にポリシー変数  $x$  があらわれるときにはその値をかえし、あらわれないときには  $y$  の値をかえす。ただし、ここで  $condition(x)$  および  $action(x)$  という関数呼びだしの値の計算は下記のステップ 1941 における方法にしたがう。そして図 22 の処理を終了する。

【0116】ステップ 1921 においては当該の定数値をもとめてかえす。たとえば、当該の定数が 1 であれば 1 という整数、"default" であれば "default" という文字列をかえす。また、当該の定数が dontCare であれば dontCare という値をかえす。そして図 22 の処理を終了する。

【0117】ステップ 1931 においては当該の局所変数の値をもとめてかえす。たとえば、当該の局所変数が Interface であれば、その値すなわち機器のインタフェース番号をかえす。そして図 22 の処理を終了する。

【0118】ステップ 1941 においては当該の関数の名称を判定する。それが condition であればステップ 1942 にすすみ、action であればステップ 1943 にすすみ、list\_number であればステップ 1944 にすすむ。ステップ 1942 においては、関数の引数としてあたえられた名称をもつ、変換中のポリシー規則の条件部にあらわれるポリシー変数の値を使用して、つぎのようにして値をもとめる。

【0119】ステップ 1941 において当該の式が  $condition(x)$  であるときには、その値はつぎのようにしてもとめる。まず  $x$  の値である文字列に ":translation" という文字列を付加してえられる文字列をキーとしてエージェント・ポリシー・データベース 1714 を検索する。その結果えられる変換方法にしたがって、変換中のポリシー規則の条件部にあらわれるポリシー変数  $x$  の値を変換した結果を  $condition(x)$  の値として使用する。たとえば  $condition(user\_class')$  の値はつぎのよう

ales' であれば、それを変数 value の値とみなす。ポリシー変数と値の対 813 によって変換方法が定義されたとき（すなわち LAN スイッチ 121 のばあい）には、変換方法  $value == Sales' ? 10.1.0.0/255.255.0.0 : value == Development' ? 10.2.0.0/255.255.0.0$  において変数 value を Sales' と置換して評価する。Sales' == Sales' は真なので評価結果は 10.1.0.0/255.255.0.0 すなわち IP アドレス 10.1.0.0 とマスク 255.255.0.0 をつなげた 8 バイトの値になる。

10 【0120】また、ステップ 1941 において当該の式が  $action(x)$  であるときには、変換中のポリシー規則の動作部にふくまれるポリシー変数  $x$  の値をそのまま  $action(x)$  の値として使用する。

【0121】たとえば、当該の式が  $condition(user\_class')$  であれば、user\_class というポリシー変数の値を使用してかえす。ただし、user\_class というポリシー変数が当該ポリシーの条件部にあらわれないときは、値が存在しないことをあらわす特殊な値 (dontCare) をかえす。そして図 22 の処理を終了する。

20 【0122】ステップ 1943 においては、関数の引数としてあたえられた名称をもつ、条件部にあらわれるポリシー変数の値をかえす。そして図 22 の処理を終了する。

【0123】ステップ 1944 においては、当該の関数呼びだしにおける第 1 引数の文字列にむすびつけられた値に 1 を加算し、その値をかえす。第 1 引数にむすびつけられた値がまだ存在しないときは、第 2 引数の値をかえし、それに 1 を加算した値を第 1 引数の文字列にむすびつける。ただし、かえすべき値が第 3 引数の値をこえているときはエラーメッセージを生成してオペレータ・コンソール 211 に送信する。たとえば、list\_number(ACL', 0, 1023) という関数呼びだしが最初に、すなわち最初のポリシー規則の変換の際によびだされるときには、値として 0 がかえされる。そして、ポリシー規則の変換のたびに list\_number(ACL', 0, 1023) がよびだされると、そのつど 1 を加算した値すなわち 1, 2, ... をかえす。その値が 1023 に達すると、エラーメッセージを生成する。すなわち、本実施例においては当該ハードウェア・テーブルサイズが 1024 であることを仮定しているの

40 を仮定しているの、当該ハードウェア・テーブルがオーバーフローするとエラーメッセージを生成する。  
【0124】本実施例においては、ユーザはオペレータ・コンソール 211 を使用して Policies ポリシー 1601 とひVariableDefinition ポリシー 701 をポリシーマネージャ 212 に配布する。また、SwitchCommands ポリシー 401 と SwitchValueTranslation ポリシー 801 とをネットワーク・エージェント 215 に配布し、ServerCommands ポリシー 501 と ServerValueTranslation ポリシー 901 とをサーバ・エージェント 216 に配布し、StorageCommands ポリシー 601 と StorageValueTranslat

ion ポリシー 1001 とをストレージ・エージェント 217 に配布する。

【0125】以上の操作をあらかじめ実行しておけば、LAN 121、サーバ 131、132、ストレージ 150、151、152 を制御するのにただ 1 個のポリシー PrioritizeMarketing1101 を配布するだけですむ。すなわち、PrioritizeMarketing 1101 を LAN スイッチ 121、サーバ 131、132 とストレージ 151、152 または ストレージ 150 に配布すればよい。3 個のストレージのうちのいずれに配布するかは、販売部門が使用するデータベースがどのストレージに存在するかによる。ただし、どこに存在するかわからないばあいには、3 個すべてに配布することも可能である。

【0126】PrioritizeMarketing 1101 の配布により、9:00 から 10:00 まで LAN においては販売部門からのトラフィックに他の部門よりたかい優先度をあたえ、サーバに関してはサーバの各リソースのなかから 50% の部分が使用できるようにし、ストレージに関しては当該のデータのキャッシングを指定することができる。また、ポリシーの変更は PrioritizeMarketing 1101 だけをかきかえればよい。

【0127】本実施例においては、あらたな機能をもつポリシーを定義するには、あらたな PolicyDefinition ポリシーと、必要なら PolicyVariableDefinition ポリシーを記述してポリシーマネージャ 212 に配布し、あらたな PolicyToCommands ポリシーと、必要なら PolicyValueTranslation ポリシーとを記述してネットワーク・エージェント 215、サーバ・エージェント 216 およびストレージ・エージェント 217 に配布すればよい。これらのポリシーの記述・変更には通常のポリシー編集インタフェースを使用すればよく、ユーザは特別な訓練を必要としないので、本発明の第 1 の目的を達することができる。

【0128】また、機器が追加されたときには、ネットワーク・エージェント 215、サーバ・エージェント 216 またはストレージ・エージェント 217 の複製をつくることをポリシーサーバ 112 に指示し、生成したあらたなエージェントに PolicyToCommands ポリシーおよび PolicyValueTranslation ポリシーを配布すればよい。これによって追加された機器に対しても既存のポリシーを容易に適用することができるため、本発明の第 2 の目的を達することができる。

【0129】以上で基本の実施例の説明をおわる。以下、前記の実施例の一部を変更した実施例について説明する。

【0130】[インタプリタへの式の入力]第 1 に、前記の基本の実施例においてはインタプリタにおいて演算子および list\_number 以外の関数の計算をおこなってなかったが、ポリシー規則において指定された値からコマンドにおいて使用される値をもとめるのに演算子や関

数を使用するべきばあいがある。たとえば、ポリシー規則において IP パケットに設定すべき DSCP (Diffserv Code Point) の値 46 がポリシー変数 DSCP によって指定されているがコマンドにおいては TOS (Type Of Service) の値 264 を指定することが必要なとき、 $4 * DSCP$  という式を使用することによってインタプリタにおいて 4 と DSCP の値の積をもとめて、これを TOS の値として使用することができる。

【0131】[list\_number 関数による CLI の検査]第 2 に、前記の基本の実施例においては list\_number 関数がハードウェア・テーブルのオーバフローの検査のために使用されていたが、list\_number 関数は CLI におけるコマンド引数値の範囲の検査のために使用することもできる。たとえば PolicyToCommands ポリシーにおいて Cisco 社のルータのための access-list コマンドを生成するばあい、access-list コマンドの最初の引数は当該コマンドによって生成すべきアクセス制御リストの番号がオペレーティング・システム IOS において厳密にきめられているために、その範囲を逸脱してはならない。たとえば、拡張 IP アクセスリストの番号は 2300 から 2699 でなければならない。このばあい、list\_number(ACL', 2300, 2699) という関数およびだしを使用することによって、きめられた範囲の番号を生成し、範囲におさまらなければいはいはエラーメッセージを生成することができる。

【0132】[定義されたポリシークラスの GUI への追加]第 3 に、前記の基本の実施例においてはオペレータ・コンソール 211 におけるエンティティの追加や削除はつねにオペレータ・コンソール 211 からポリシーマネージャ 212 への要求というかたちで実行されていた。そのため、PolicyDefinition ポリシーがポリシーマネージャ 212 に配布されることによってあらたなポリシークラスが定義されても、それはポリシークラス・リスト 312 には表示されなかった。ポリシークラス・リスト 312 の表示をポリシーマネージャ 212 において定義されたポリシークラスのリストと一致させるためには、ポリシークラスがポリシーマネージャ 212 において定義または削除されたときに、それをオペレータ・コンソール 211 に通知するインタフェースを追加すればよい。すなわち、PolicyDefinition ポリシーがマネージャ制御部 1212 において定義または削除されたとき、すなわちステップ 1345 および 1346 の直後において、上位受信部 1211 をつうじてオペレータ・コンソール 211 に create(name) または deleted(name) という通知をおこなえばよい。ここで created および deleted が通知の名称であり、name は定義または削除されたポリシークラス名である。オペレータ・コンソール 211 はこれらの通知を受けとったときに当該のポリシークラス名をポリシークラス・リスト 312 に追加またはポリシークラス・リスト 312 から削除すればよい。

【0133】このようなポリシーマネージャ 212 からの通知にもとづくエンティティ名の追加・削除はポリシークラス名だけでなくポリシー名、ポリシー規則名の追加・削除にも使用することができる。それにより、オペレータ・コンソール 211 以外からエンティティ追加・削除されても、その結果をオペレータ・コンソール 211 に反映させることができる。

【0134】[GUI 上での構文検査]第 4 に、前記の基本の実施例においてはオペレータ・コンソール 211 から入力されたポリシーにふくまれる不正なポリシー変数名の使用やポリシー変数とその値の型の不一致などのエラーは、ポリシーマネージャ 212 において検出されてオペレータ・コンソール 211 に通知されていた。しかし、入力されたエンティティをポリシーマネージャ 212 に送付するまえに、エラーをふくむ表現をただちに検出することがのぞましい。そのためにはつぎのような方法をとることができる。オペレータ・コンソール 211 においてポリシー規則を入力する際には、事前に当該のポリシー規則が属するポリシーのポリシークラス名を入力するようにし、またポリシー規則の条件部および動作部の入力においてはポリシー変数および値を単位として入力するようにする。

【0135】ポリシー変数が入力されたときには is\_valid\_variable(class\_name, condition\_or\_action, variable\_name) という要求インタフェースをつうじてオペレータ・コンソール 211 からポリシーマネージャ 212 にといいわせをおこなう。ここで class\_name は当該のポリシークラス名であり、condition\_or\_action は条件部か動作部かというくべつをあらわし、variable\_name は当該のポリシー変数名である。ポリシーマネージャ 212 はこの要求をうけるとサーバポリシー・データベース 1214 を検索して当該のポリシー変数が指定された条件部または動作部に出現可能かどうかをしらべ、その結果をオペレータ・コンソール 211 にかえす。出現可能でないばあいには、オペレータ・コンソール 211 はエラーメッセージを表示してユーザの入力をとりけす。

【0136】[COPS-PR によるコマンド送付]第 5 に、前記の基本の実施例においてはネットワーク・エージェント 215 において LAN スイッチ 212 のハードウェア・テーブルへの設定をおこなった。しかし、おおくのネットワーク機器はコマンドライン・インタフェース、プロトコル SNMP によって設定すべき MIB、プロトコル COPS-PR によって設定すべき PIB (Policy Information Base) などのインタフェースをもっている。したがって、ネットワーク機器への設定のためのポリシーエージェントにおいてこれらのインタフェースを使用することができる。COPS-PR を使用するばあいには、SNMP における set コマンドのかわりに decision (install/remove) コマンドを使用すればよい。

【0137】[XML の使用]第 6 に、前記の基本の実施

例においてはポリシーを入力するインタフェースは GUI だけだったが、ポリシー入力のために XML を使用することができる。すなわち、ポリシー制御部 1212 はポリシーを上位受信部 1211 から入力するかわりにポリシーを記述した XML テキストを XML パーザをとおして入力することができる。ポリシーを XML テキストによって表現することができることはあきらかだが、たとえば PolicyDefinition ポリシー 1601 をつぎのような XML テキストとして記述することができる。

```
<policy>
<name>Policies</name>
<class>PolicyDefinition</class>
<conditions>
<condition><variable>name</variable><value>QoS</value></condition>
</conditions>
<actions>
<action>
<variable>condition_variables</variable>
<value><list><element>user_class</element></list></value>
</action>
<action>
<variable>action_variables</variable>
<value><list><element>access</element></list></value>
</action>
</actions>
</policy>
```

ここで <policy>…</policy> は “…” の部分がポリシーであることをあらわすタグであり、<name>…</name> は “…” の部分がなまえであることをあらわすタグであり、<class>…</class> は “…” がポリシークラスであることをあらわすタグである。<conditions>…</conditions> は “…” が条件部であることをあらわすタグであり、<condition>…</condition> は “…” が 1 個の条件であることをあらわすタグであり、<variable>…</variable> は “…” が変数名であることをあらわすタグであり、<value>…</value> は “…” が変数の値であることをあらわすタグである。<actions>…</actions> は “…” が動作部であることをあらわすタグであり、<action>…</action> は “…” が 1 個の動作であることをあらわすタグであり、<list>…</list> は当該の値がリストであることをあらわすタグであり、<element>…</element> は “…” が当該リストの要素であることをあらわすタグである。

【0138】上記の XML テキストは任意のポリシーを表現できる形式にしたがっている。これにより、前記の基本の実施例におけるのと同様に、通常のポリシーを入力するのと同じの方法にしたがってポリシー定義および

ポリシー変換法の定義を入力することができる。しかし、ポリシー定義、ポリシー変換法の定義のための専用のタグを使用することもできる。たとえば PolicyDefinition ポリシー 1601 をつぎのような XML テキストとして記述することができる。

```
<policy>
<name>Policies</name>
<class>PolicyDefinition</class>
<polycname>QoS</polycname>
<condition_variables><variable>user_class</variable>
</condition_variables>
<action_variables><variable>access</variable></action_variables>
</policy>
```

ここで <polycname>...</polycname> は "... " が定義すべきポリシー名であることをあらわし、<condition\_variables>...</condition\_variables> は "... " が条件部にあらわれる変数のリストであることをあらわし、<action\_variables>...</action\_variables> は "... " が動作部にあらわれる変数のリストであることをあらわして

【0139】

【発明の効果】本発明のポリシーシステムを使用すれば、既存のインタフェースを再利用して容易に必要な応じてポリシーの追加や削除ができる。また、あらたな機器インタフェースやコマンドをもつあらたな機器に、現場における短時間の開発によってポリシーを必要に応じて適用できるようにすることができる。

【図面の簡単な説明】

【図 1】本実施例におけるネットワーク構成図である。

【図 2】図 1 におけるポリシーサーバ 112 の構成図である。

【図 3】図 2 におけるオペレータ・コンソール 211 の GUI の説明図である。

【図 4】ポリシーサーバ 112 に入力するポリシー SwitchCommands の説明図である。

【図 5】ポリシーサーバ 112 に入力するポリシー Serv

erCommands の説明図である。

【図 6】ポリシーサーバ 112 に入力するポリシー StorageCommands の説明図である。

【図 7】ポリシーサーバ 112 に入力するポリシー VariableDefinition の説明図である。

【図 8】ポリシーサーバ 112 に入力するポリシー SwitchValueTranslation の説明図である。

【図 9】ポリシーサーバ 112 に入力するポリシー ServerValueTranslation の説明図である。

【図 10】ポリシーサーバ 112 に入力するポリシー StorageValueTranslation の説明図である。

【図 11】ポリシーサーバ 112 に入力するポリシー PrioritizeMarketing の説明図である。

【図 12】図 2 におけるポリシーマネージャ 212 の構成図である。

【図 13】図 12 のポリシーマネージャ 212 の処理フロー説明図である。

【図 14】図 12 のポリシーマネージャ 212 におけるサーバポリシー・コンパイラ 1213 の処理フロー説明図である。

【図 15】図 12 のポリシーマネージャ 212 におけるポリシー検査部 1215 の処理フロー説明図である。

【図 16】ポリシーサーバ 112 に入力するポリシー Policies 1601 の説明図である。

【図 17】図 2 におけるポリシーエージェント 215、216、217 の構成図である。

【図 18】図 17 のポリシーエージェント 1701 の処理フロー説明図である。

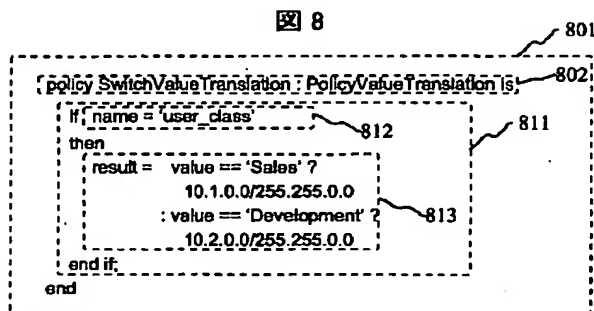
【図 19】図 17 のポリシーエージェント 1701 におけるエージェント・ポリシー・コンパイラ 1713 の処理フロー説明図である。

【図 20】図 17 のポリシーエージェント 1701 におけるポリシー変換部 1715 の処理フロー説明図である。

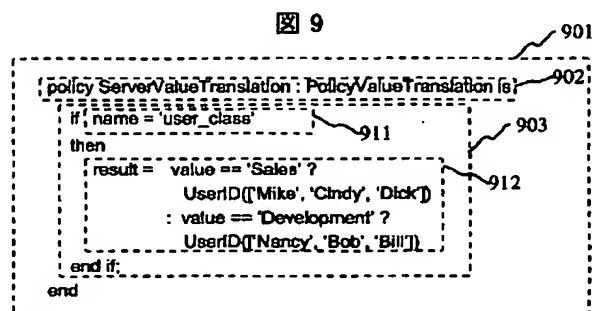
【図 21】図 17 のポリシーエージェント 1701 におけるコマンド変換部 1716 の処理フロー説明図である。

【図 22】図 17 のポリシーエージェント 1701 におけるインタプリタ 1717 の処理フロー説明図である。

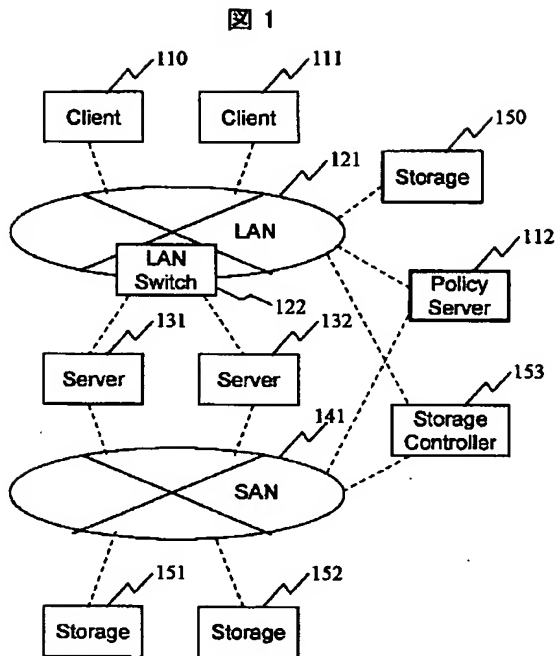
【図 8】



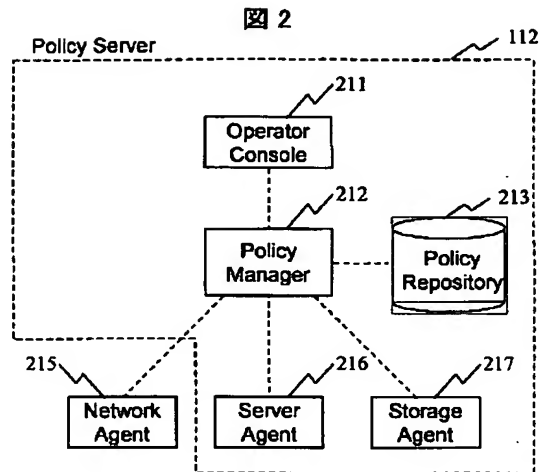
【図 9】



【図 1】

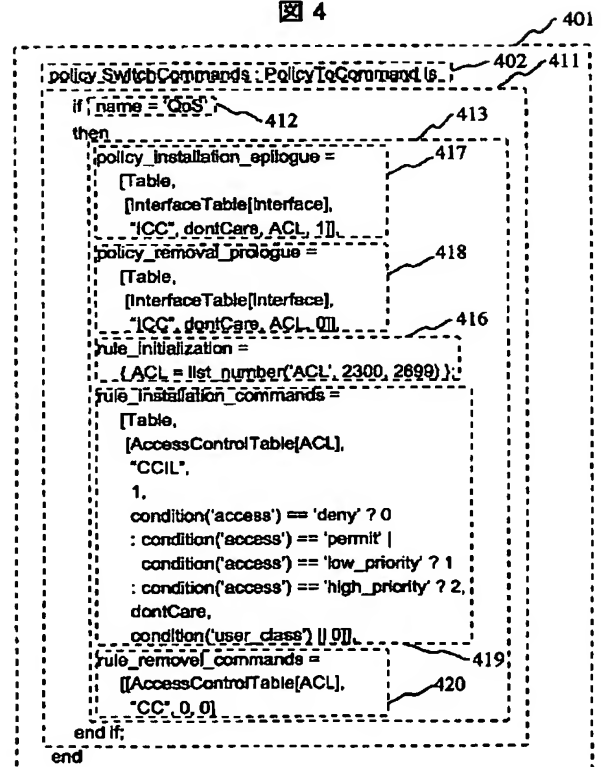


【図 2】

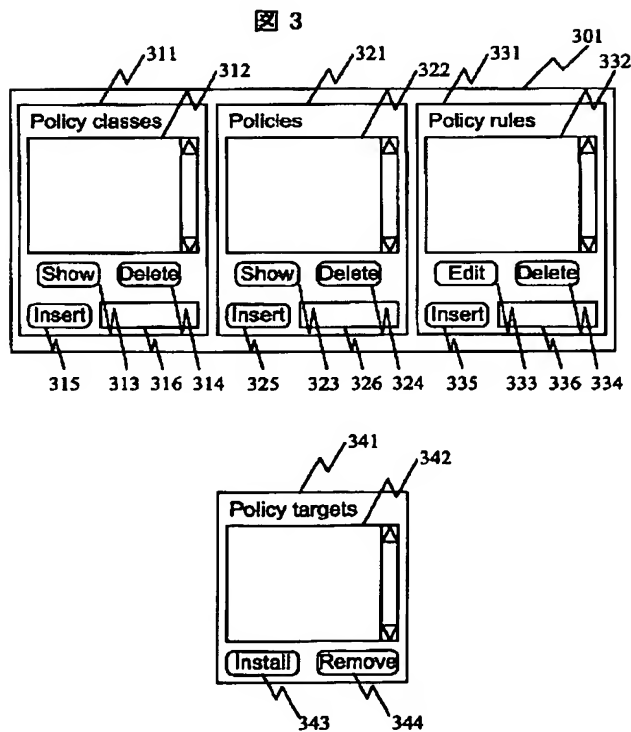


【図 4】

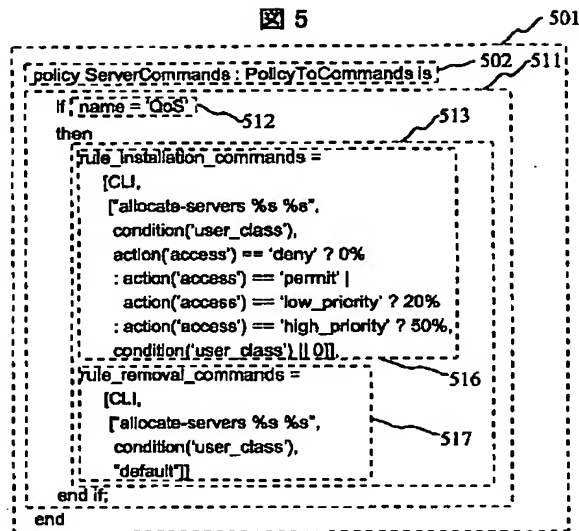
図 4



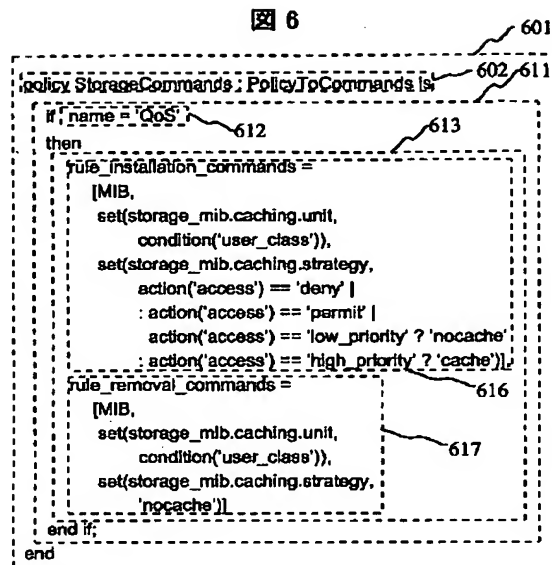
【図 3】



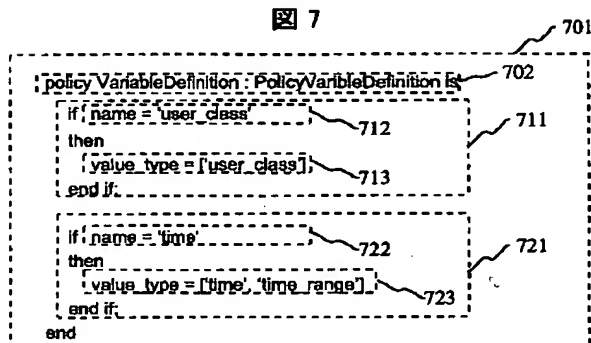
【図 5】



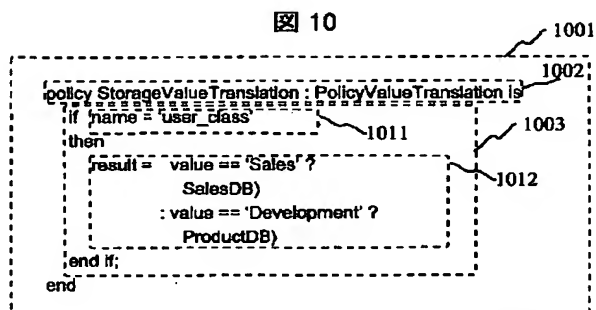
【図 6】



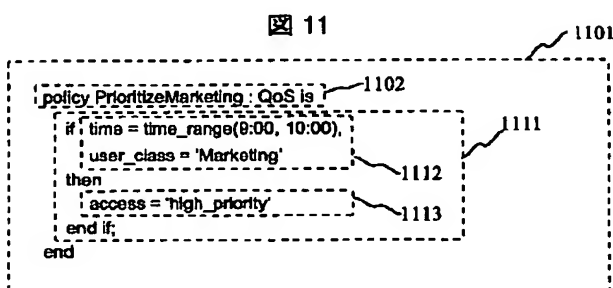
【図 7】



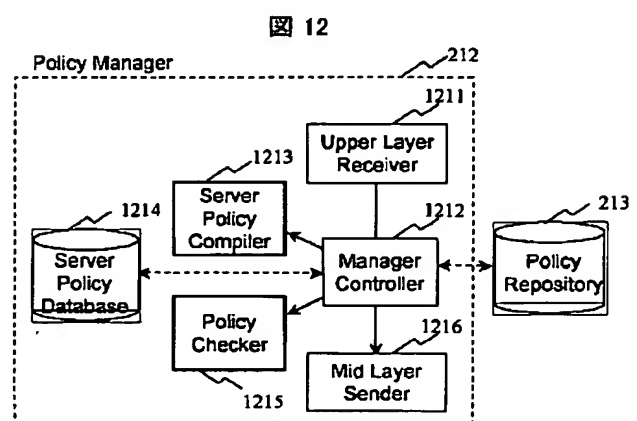
【図 10】



【図 11】

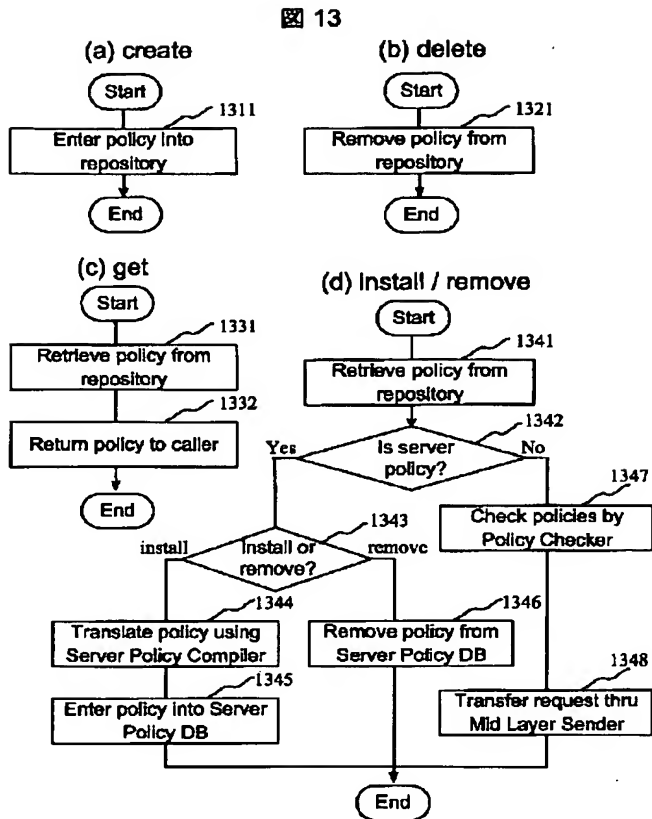


【図 12】

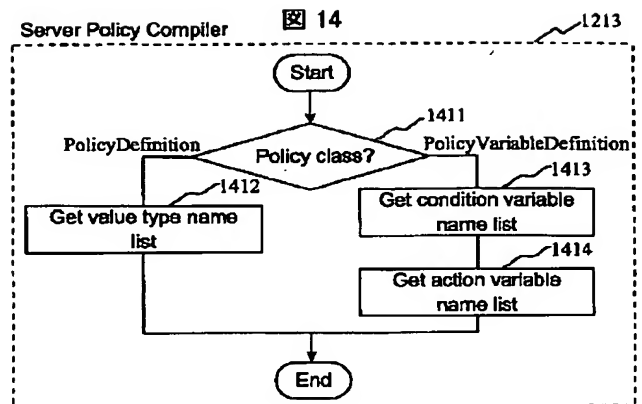




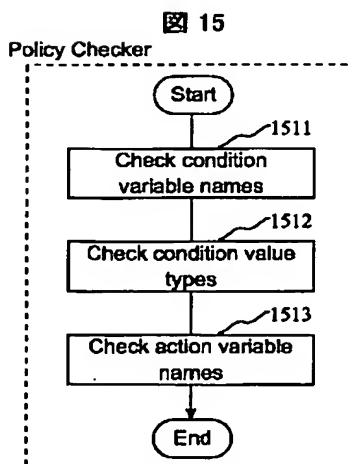
【図 13】



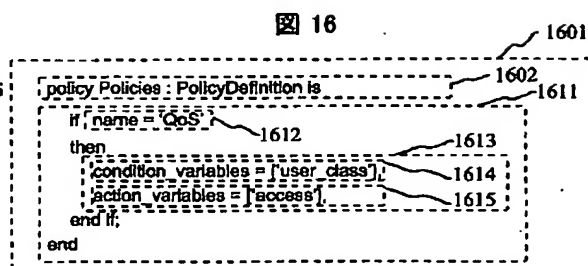
【図 14】



【図 15】

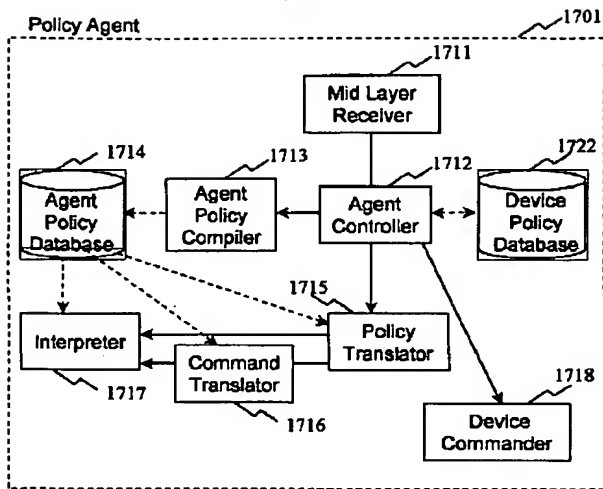


【図 16】



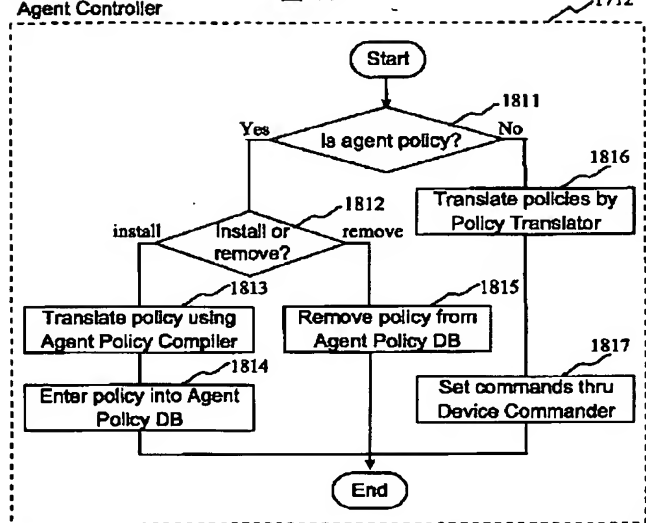
【図 17】

図 17



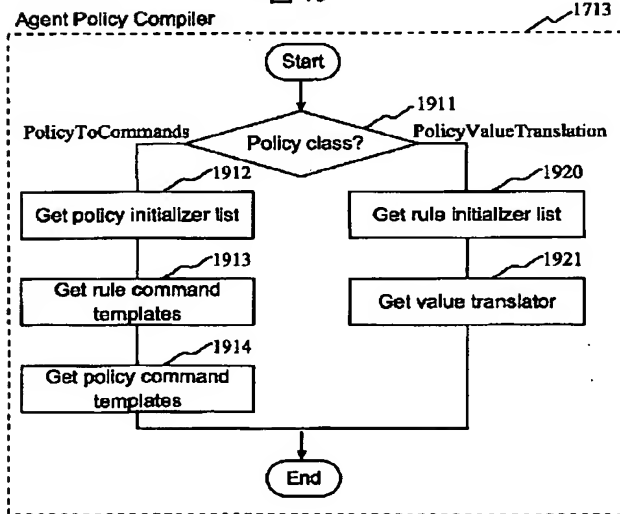
【図 18】

図 18



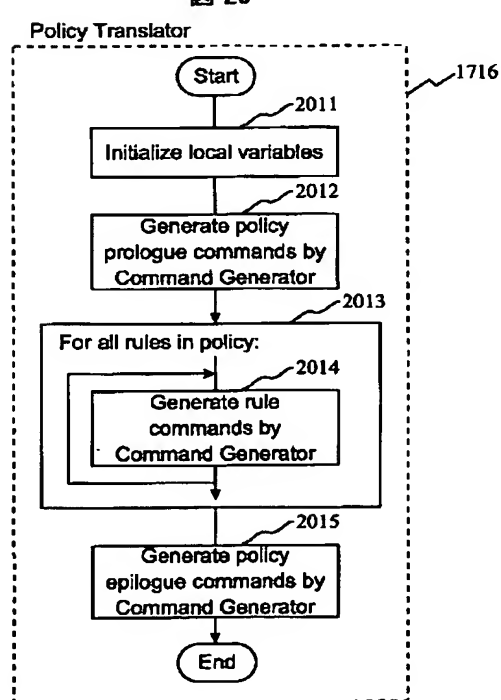
【図 19】

図 19

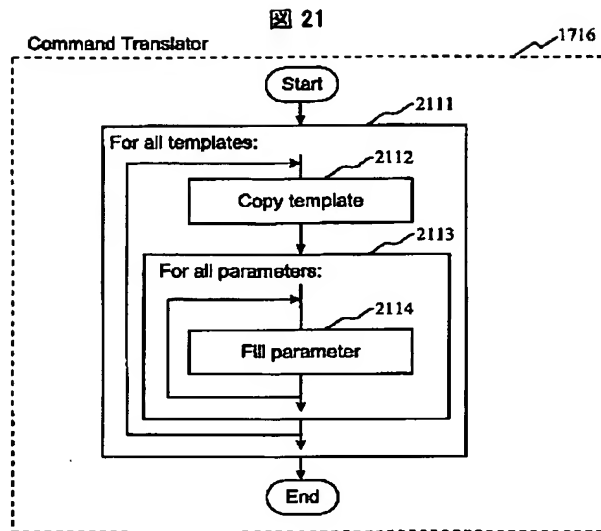


【図 20】

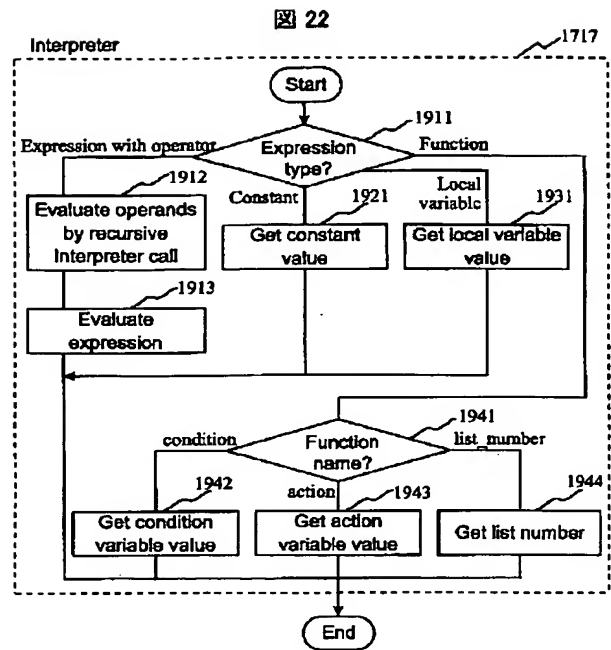
図 20



【図 21】



【図 22】



フロントページの続き

Fターム(参考) 5B085 AC03  
 5B089 GA11 JA35 JB10 KA09 KA10  
 KB03 KF05  
 5K030 GA11 JA10 KA02 KA07 MD07